

# BPMLight - NewUI

## Studienarbeit

Abteilung Informatik  
Hochschule für Technik Rapperswil

Herbstsemester 2012

Autor(en): Bejtulla Akiku  
Betreuer: Prof. Hansjörg Huser  
Projektpartner: bbv ICT Solutions AG, Zug

## Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.
- dass ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum:

Name, Unterschrift:

## MetroStyle

### Aufgabenstellung für Akiku Bejtulla ( [REDACTED] )

Aufgabe: gemäss Aufgabenstellung von Adrian Bachofen

Resultate: gemäss Aufgabenstellung von Adrian Bachofen, Dokumente gemäss Vorgaben

Ziele:

- Qualifizierter Feedback zu aktuellem UI der SharePoint Implementation (Look/Feel, weniger zur Navigation da noch nicht genügend aufgebaut)
- Vorschlag/Varianten des MetroStyle Look and Feel: Hauptfokus: Liste der Prozessinstanzen und Darstellung einer Prozessinstanz
- Prototype des gewählten UI Vorschlages auf einem Windows Surface Tablet

Erfolgsdefinition:

- Cooles Aussehen / Moderne Oberfläche

### Auftraggeber

Adrian Bachofen

CEO bbv ICT Solutions AG



### Projektteam

Bejtulla Akiku ( [REDACTED] )

### Betreuung HSR

Hansjörg Huser, [REDACTED]

## Projektabwicklung

### Termine:

Siehe auch Terminplan auf <https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html?&L=0>

- 17.09.12: Beginn der Studienarbeit, Ausgabe der Aufgabenstellung.
- 17.12.12: Die Studierenden senden folgende Dokumente der Arbeit per Email zur Prüfung an ihre Betreuer: - Kurzfassung - A0-Poster  
Vorlagen sowie eine ausführliche Anleitung betreffend Dokumentation stehen unter den allgemeinen Infos Diplom-, Bachelor- und Studienarbeiten zur Verfügung.
- 21.12.12: Die Studierenden senden die vom Betreuer abgenommene und freigegebene Kurzfassung als Word-Dokument an das Studiengangsekretariat (cfurrer(at)hsr.ch).
- 21.12.12: Abgabe des Berichtes an den Betreuer bis 17.00 Uhr.
- Zwischenbesprechung/Review mit Auftraggeber nach Projektplan

## Arbeitsaufwand

Für die erfolgreich abgeschlossene Arbeit werden 8 ECTS angerechnet. Dies entspricht einer Arbeitsleistung von mind. 240 Stunden pro Student.

## Hinweise für die Gliederung und Abwicklung des Projektes:

Gliedern Sie Ihre Arbeit in 4 bis 5 Teilschritte. Schliessen Sie jeden Teilschritt mit einem Meilenstein ab. Definieren Sie für jeden Meilenstein, welche Resultate dann vorliegen müssen!

Folgende Teilschritte bzw. Meilensteine sollten Sie in Ihrer Planung vorsehen:

- Schritt 1: Projektauftrag inkl. Projektplan (mit Meilensteinen),
  - Meilenstein 1: Review des Projektauftrages abgeschlossen. Projektauftrag von Auftraggeber und Dozent genehmigt
  - Letzter Meilenstein: Systemtest abgeschlossen
  - Termin: ca. eine Woche vor Abgabe
- Entwickeln Sie Ihre SW mit einem agilen oder in einem iterativen, inkrementellen Prozess: Planen Sie möglichst früh einen ersten lauffähigen Prototypen mit den wichtigsten und kritischsten Kernfunktionen. In die folgenden Phasen können Sie dieses Kernsystem schrittweise ausbauen und testen.
- Falls Sie in Ihrer Arbeit neue oder Ihnen unbekannte Technologien einsetzen, sollten Sie parallel zum Erarbeiten des Projektauftrages mit dem Technologiestudium beginnen.
- Setzen Sie konsequent Unit-Tests ein! Verwalten Sie ihre Software und Dokumente in einem Versionsverwaltungssystem (SVN, TFS). Stellen Sie sicher, dass der/die Betreuer jederzeit Zugriff auf das Repository haben und dass das Projekt anhand des Repositories jederzeit wiederhergestellt werden kann.
- Achten Sie auf die Einhaltung guter Programmier- und Designprinzipien
- Halten Sie sich im Übrigen an die Vorgaben aus dem Modul SE-Projekt.

## Projektadministration

- Führen Sie ein individuelles Projekttagbuch aus dem ersichtlich wird, welche Arbeiten Sie durchgeführt haben (inkl. Zeitaufwand). Diese Angaben sollten u.a. eine individuelle Beurteilung ermöglichen.
- Dokumentieren Sie Ihre Arbeiten laufend. Legen Sie Ihre Projektdokumentation mit der aktuellen Planung und den Beschreibungen der Arbeitsergebnisse elektronisch in einem Projektordner ab. Dieser Projektordner sollte jederzeit einsehbar sein (z.B. svn-Server oder File-Share).

## Inhalt der Dokumentation

Bei der Abgabe muss jede Arbeit folgende Inhalte haben:

- Dokumente gemäss Vorgabe:
  - siehe <https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html?&L=0>.
- Die Abgabe ist so zu gliedern, dass die obigen Inhalte klar erkenntlich und auffindbar sind.
- Zitate sind zu kennzeichnen, die Quelle ist anzugeben.
- Verwendete Dokumente und Literatur sind in einem Literaturverzeichnis aufzuführen.
- Projekttagbuch, Dokumentation des Projektverlaufes, Planung etc.

Die Dokumentation (inkl. Source) ist vollständig auf CD/DVD in 3 Exemplaren abzugeben. Auf Wunsch ist für den Betreuer eine gedruckte Version zu erstellen.

## Fortschrittsbesprechung:

Regelmässig findet zu einem fixen Zeitpunkt eine Fortschrittsbesprechung statt.

## Inhaltsverzeichnis

Abstract .....	iv
Management Summary.....	v
Ausgangslage .....	v
Vorgehen .....	v
Ergebnisse .....	v
Ausblick .....	v
<i>Technischer Bericht</i> .....	1
1. Einleitung.....	2
1.1 Aufgabenstellung.....	2
1.2 Architekturübersicht .....	2
2. Ergebnisse.....	3
2.1 Evaluierung der bestehenden SharePoint-Webseite .....	3
2.2 Von SharePoint zu Windows Store App .....	4
2.3 Evaluation Windows Store Apps (Metro UI) .....	6
2.3.1 Windows Store .....	6
2.3.2 Musik Applikation.....	6
2.3.3 Mockups .....	7
2.4 Vorschlag endgültige Benutzeroberfläche .....	10
2.4.1 Startseite .....	10
2.4.2 Prozesse.....	13
2.4.3 Prozessinstanzen .....	14
2.4.4 Detailansicht.....	14
2.4.5 Favoriten.....	15
2.4.6 Checkliste quittieren .....	16
2.5 Prototyp.....	17
2.5.1 Einschränkungen .....	17
2.5.2 Erweiterungen .....	17
2.5.3 Entwicklungsumgebung .....	18
2.5.4 Wiederverwendeter Code .....	18
2.5.5 Mock Web Service .....	19
2.5.6 Windows Store App.....	21
2.5.7 Administration.....	28
2.5.8 NotificationExtensions.....	28
3. Schlussfolgerung.....	29
3.1 Zusammenfassung.....	29

3.2 Beurteilung der Resultate.....	29
4. Persönlicher Bericht .....	29
5. Literaturverzeichnis.....	31
6. Glossar und Abkürzungsverzeichnis .....	32
7. Abbildungsverzeichnis.....	33
8. Skizzenverzeichnis .....	34
9. Tabellenverzeichnis .....	34
10. Mockupsverzeichnis .....	34
<i> Projektdokumentation</i> .....	35
11. Projektplan.docx.....	36
11.1 Einführung .....	37
11.1.1 Zweck.....	37
11.1.2 Gültigkeitsbereich.....	37
11.1.3 Referenzen .....	37
11.1.4 Übersicht .....	37
11.2 Projekt Übersicht.....	38
11.2.1 Zweck und Ziel .....	38
11.2.2 Annahmen und Einschränkungen .....	38
11.2.3 Kein System.Data Namespace – Keine lokale Datenbank .....	38
11.2.4 Kein WSHttBinding – Kein SOAP 1.2.....	38
11.4 Projektorganisation .....	39
11.4.1 Organisationsstruktur.....	39
11.4.2 Externe Schnittstellen .....	39
11.5 Management Abläufe.....	40
11.5.1 Projekt Kostenvoranschlag.....	40
11.5.2 Projektplan / Zeitplan.....	40
11.5.3 Iterationsplanung .....	40
11.5.4 Meilensteine.....	40
11.5.5 Besprechungen.....	41
11.5.6 Abgaben.....	41
11.6 Arbeitspakete .....	42
11.6.1 User Interface .....	42
11.6.2 Use Case .....	42
11.6.3 Prototyp.....	43
11.7 Infrastruktur .....	44
11.8 Qualitätsmassnahmen.....	45
12. Projektplan.xlsx .....	46
13. Mockup.docx .....	51

---

13.1 Einführung .....	52
13.2 Erster Entwurf .....	52
13.2.1 Metro Beispiele .....	53
13.2.2 Resultat .....	55
13.3 Zweiter Entwurf.....	55
13.3.1 Variante 1 .....	56
13.3.2 Variante 1.5 .....	57
13.3.3 Variante 1.75 .....	57
13.3.4 Variante 2 .....	61
13.3.5 Variante 3 .....	63
13.3.6 Resultat.....	64
13.4 Dritter Entwurf .....	65
13.4.1 Resultat.....	66
13.5 Vierter Entwurf.....	66
13.5.1 Resultat.....	68
13.6 Fünfter Entwurf .....	69
13.6.1 Startseite .....	70
13.6.2 Prozesse.....	72
13.6.3 Prozessinstanzen .....	73
13.6.4 Detailansicht.....	74
13.6.5 Resultat.....	77

## Abstract

Die bbv ICT Solutions AG betreibt für interne Zwecke eine SharePoint Webseite, die der Erstellung und Beobachtung von Workflows dient. Dabei können Prozesse definiert und Instanzen davon erstellt werden. Zu jeder Prozessinstanz können dazugehörige Dokumente und Checklisten zugeschrieben werden, welche bei Erledigung von einer zuständigen Person quittiert werden.

Die Aufgabe dieser Studienarbeit beinhaltete, diese SharePoint Webseite zu bewerten und einen Vorschlag für eine moderne Tablet Applikation auf Basis von Windows 8 und dem Microsoft Surface RT Tablet vorzustellen. Als Quelle der Daten dient ein SOAP1.1 Web Service.

Aufgrund des Arbeitsaufwandes können nicht alle Funktionen der SharePoint Webseite im Prototypen der Windows Store App (offizielle Bezeichnung, früher als Metro-Style oder NewUI bekannt) berücksichtigt werden.

Der Fokus lag deshalb auf den Einstieg in die Applikation, die Darstellung der Prozessen und Instanzen sowie die Darstellung der jeweiligen Checklisten.

Der zu erstellende Prototyp ist auf Benutzer aus der Führungsebene ausgerichtet und soll als effizientes und einfaches Werkzeug dienen. Es ist deswegen eine Funktion zu finden, welche in der ursprünglichen SharePoint Webseite gar nicht vorhanden ist: Benutzer können Ihre eigenen Favoriten erstellen, um so Ihre Arbeit zu beschleunigen und auch gerne die App verwenden.

## Management Summary

### Ausgangslage

Jedes Unternehmen ist darauf angewiesen ihre Arbeitsabläufe zu verwalten, um Vorteile daraus zu schlagen, Optimierungen durchzuführen oder einen Überblick zu bekommen. Die bbv ICT Solutions AG verwendet dafür eine Web Seite basierend auf einen SharePoint-Server.

Auf dieser Plattform können die Benutzer Prozesse und Prozessinstanzen erstellen. Eine Prozessinstanz zeichnet sich daraus aus, dass ihr verschiedene Dokumente und Checklisten zugewiesen werden. Diese Checklisten werden stets bearbeitet bis sie abgeschlossen werden.

Diese Studienarbeit soll dazu dienen, die Möglichkeit zu erkunden, wie die Funktionalitäten der SharePoint-Web Seite auf einem Microsoft Surface RT Tablet implementiert werden können. Dazu sollen Vorschläge gemacht werden und schliesslich ein Prototyp, also eine Windows Store App, programmiert werden.

### Vorgehen

Um sich eine besseres Bild über die Anforderungen der Applikation zu machen, wurde die bestehende SharePoint-Web Seite ausprobiert.

Zeitgleich musste man sich in die Bedienung und die Philosophie der neuen Benutzeroberfläche des gewünschten Ziel-Tablet einarbeiten.

Aus diesen Erfahrungen konnten schon die ersten Entwürfe und Vorschläge erarbeitet werden, die in ständiger Absprache mit dem Auftraggeber immer weiter verfeinert wurden.

### Ergebnisse

Dem Auftraggeber kann ein Dokument überliefert werden, worin die Evolution der Vorschläge und Bedeutungen der darin enthaltenden Elemente der Benutzeroberfläche erläutert werden.

Für den akzeptierten Vorschlag der Benutzeroberfläche steht ein lauffähiges Programm zur Verfügung.

### Ausblick

Mit diesen Ergebnissen kann der Auftraggeber eine Entscheidung fällen, ob diese Windows Store App so weitergeführt werden kann oder ein anderer Ansatz gewählt wird. Zudem sind mit dem Prototyp schon einige Stolpersteine aus dem Weg geräumt und man hat einen Einblick in die Entwicklung von Windows Store Apps bekommen.

# *Technischer Bericht*

## 1. Einleitung

### 1.1 Aufgabenstellung

In dieser Studienarbeit ging es darum, eine SharePoint-Webseite, mit der man Workflows erstellen und verwalten kann, nach dem Look & Feel zu beurteilen und um später eine moderne Tablet-Applikation zu entwickeln, die eingeschränkte Möglichkeiten der Webseite bietet. Die Applikation soll prinzipiell an den Benutzer „Controller“ ausgerichtet sein. Ein Controller ist ein Benutzer, der den Überblick über seine zugewiesenen Prozessinstanzen hat und diese überwacht.

Es sollen zuerst mehrere Vorschläge und Varianten für die Benutzeroberfläche auf Basis der neuen Windows-Oberfläche, auch bekannt als Metro UI, erarbeitet und evaluiert werden. Der Prototyp dieser Tablet-Applikation sollte auf dem Microsoft Surface Tablet mit Windows 8 als Betriebssystem abgeschnitten sein.

Neben der SharePoint-Webseite steht ein SOAP v1.2 Web Service zur Verfügung welcher von der Tablet-Applikation konsumiert werden soll.

### 1.2 Architekturübersicht

Für den Prototyp und diese Arbeit relevante Architektur ist sehr simple. Das Tablet greift auf einen Web Service zu und verwertet jene Antworten.



**Abbildung 1 Architekturübersicht**

Da jedoch der Web Service vom Typ SOAP v1.2 nicht kompatibel mit Windows Store Apps (offizielle Bezeichnung von den modernen Windows Applikation mit der neuen Benutzeroberfläche) ist, wurde für die Entwicklung ein eigens erstellter Web Service verwendet.

Im Folgenden werden die verwendeten Methoden des Web Service kurz beschrieben:

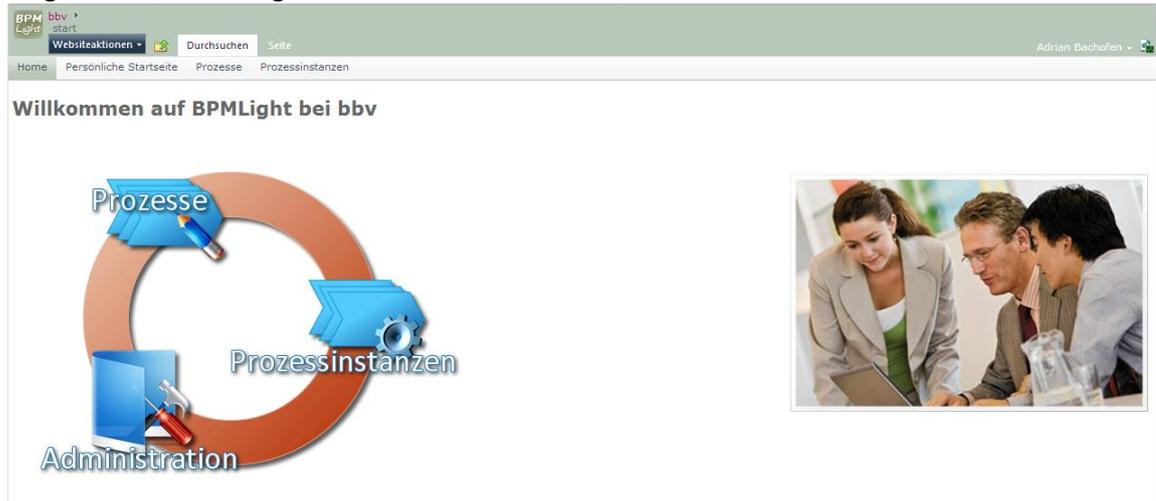
Methode	Kurzbeschreibung
ListProcesses	Liefert eine Menge von Prozessen zurück
ListInstances	Liefert eine Menge von Prozessinstanzen zurück
ListInstancesByProcess	Liefert alle einem Prozess zugehörigen Prozessinstanzen zurück
ListSingleInstanceById	Liefert eine einzige Instanz zu einem verlangten Prozess
ListChecklistByProcessInstance	Liefert eine Menge von Checklisten einer zugehörigen Prozessinstanz zurück
ListChecklistByStatus	Liefert eine Menge von Checklisten eines angegebenen Status zurück

**Tabelle 1 Web Service Methoden**

## 2. Ergebnisse

### 2.1 Evaluierung der bestehenden SharePoint-Webseite

Folgend ist der Einstieg der SharePoint-Webseite zu sehen:



**Abbildung 2 Startseite SharePoint**

Hier ist der Fokus der Seite nicht klar erkennbar, obwohl nur hinter dem Ring mit den drei Elementen auf der linken Seite Funktionen hinterlegt sind. Somit sollte der Ring mehr in den Fokus des Benutzers rücken.

Die Schaltfläche „Prozesse“ aus dem Ring führt zu einer Auflistung von allen Prozessen.

Leider ist in folgender Abbildung nur ein Prozess sichtbar:

Prozesse		
ID	Prozessname	Instanz erstellen
17	Personaleintritt	<a href="https://test.bpmlight.ch/bbv/BPMLight/Processes.aspx?create=17">https://test.bpmlight.ch/bbv/BPMLight/Processes.aspx?create=17</a>

[+ Prozess hinzufügen](#)

**Abbildung 3 Liste von Prozessen**

Der obigen Abbildung entsprechend sieht auch die Liste von allen vorhandenen Prozessinstanzen aus.

In der nächsten Abbildung ist die Detail-Ansicht einer Prozessinstanz zu sehen:

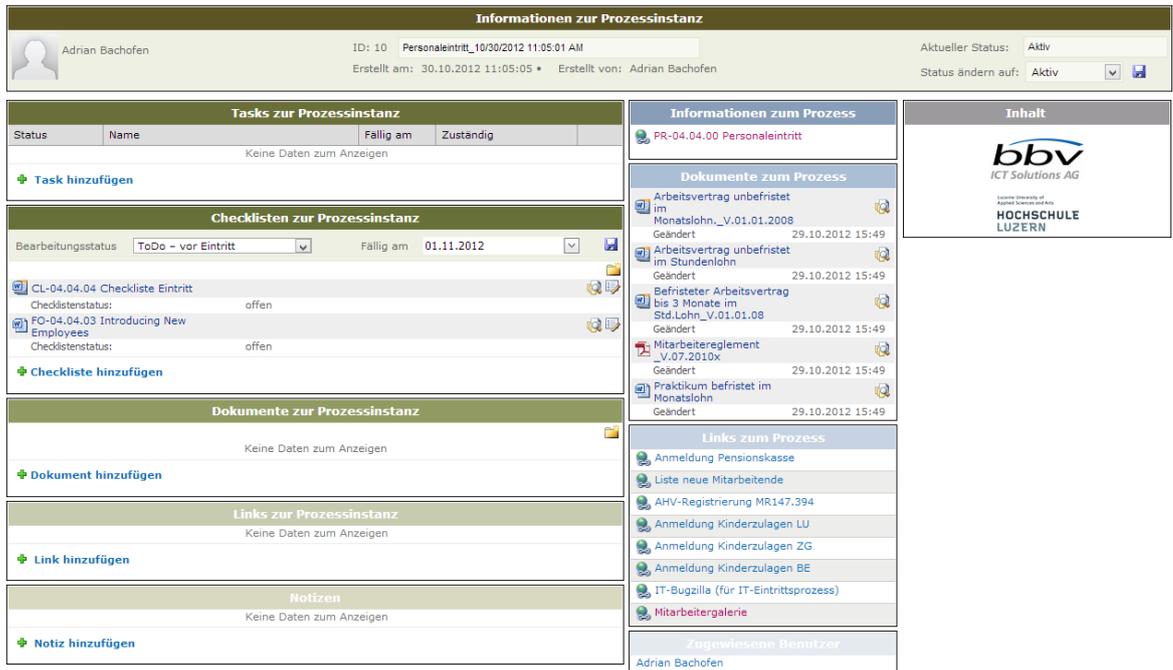


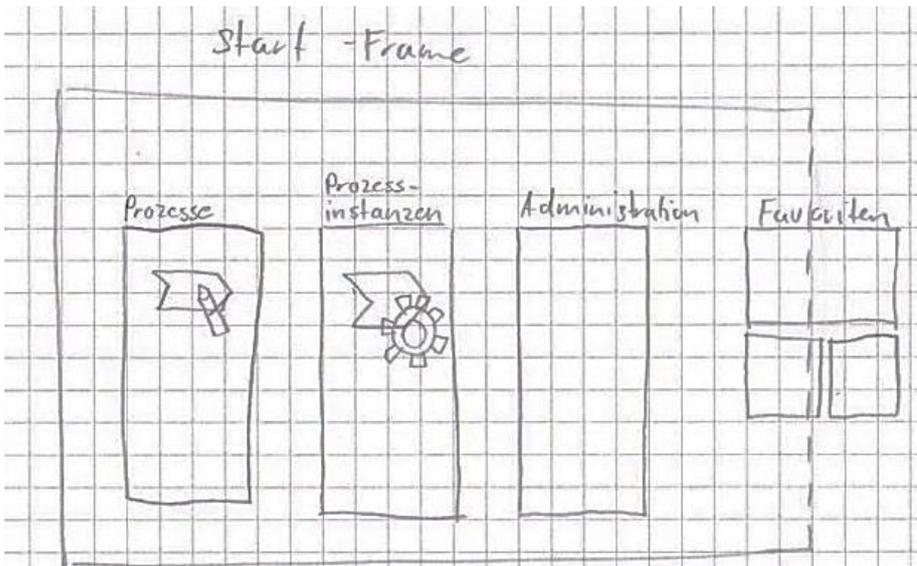
Abbildung 4 Detail-Ansicht einer Prozessinstanz

Hier wurde, unter anderem, folgende Sachen angemerkt:

- Die Elemente (Tasks, Checklisten, Dokumente, Links und Notizen) in der linken Spalte die keinen Inhalt anzeigen, sollte in den Hintergrund treten. Diese können zum Beispiel ausgegraut werden. Der Fokus des Benutzers soll stets mit dem relevanten beansprucht werden.
- Bei „Zugewiesene Benutzer“ kann sogleich der Avatar des Benutzers verwendet werden.

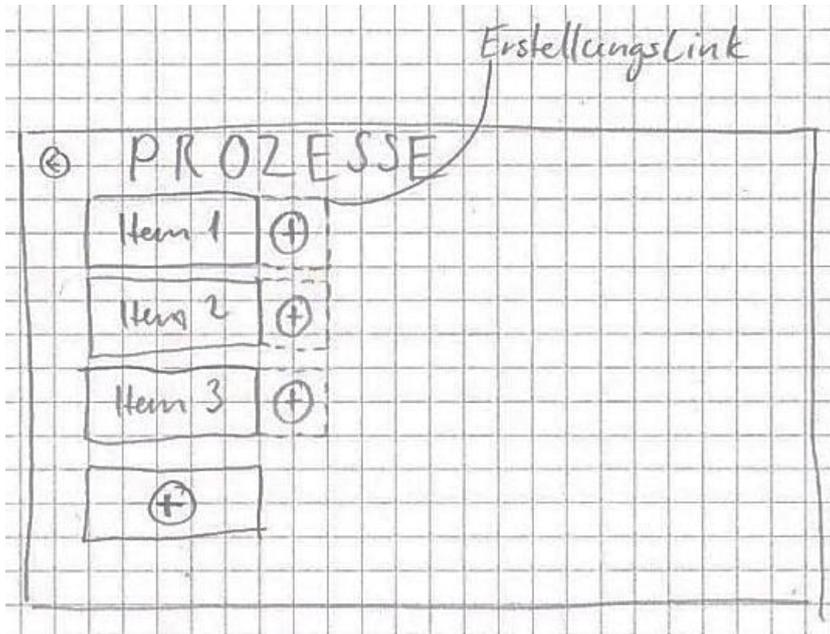
## 2.2 Von SharePoint zu Windows Store App

Die erste Idee war, eine Tablet-App analog zu der SharePoint Webseite mit ihrem ganzen Funktionsumfang zu skizzieren. Diese Überlegung führte zu diesen Skizzen:



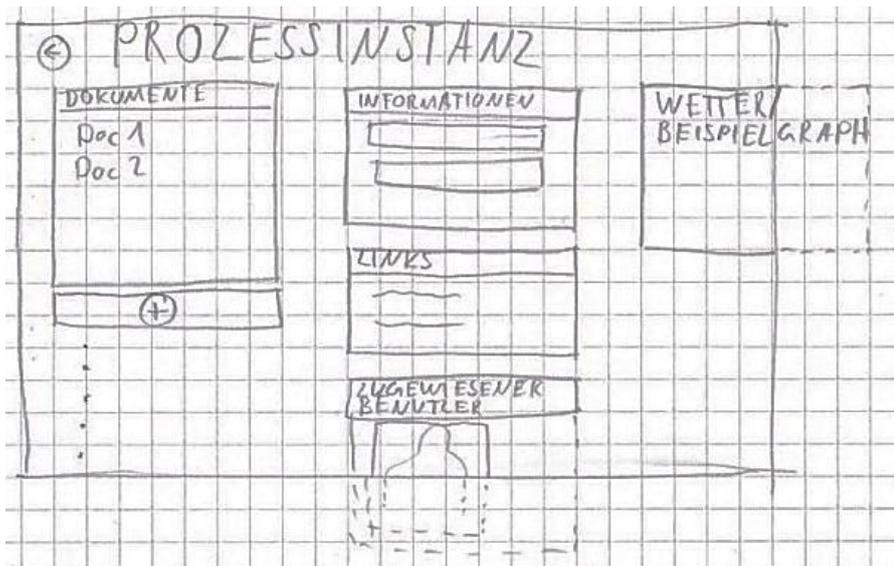
Skizze 1 Startseite

- Der Start-Frame enthält grundsätzlich die gleiche Funktionalität wie die Startseite der SharePoint Webseite. Man kann von hier aus nach ‚Prozesse‘, ‚Prozessinstanzen‘ und ‚Administration‘ navigieren. Zusätzlich können hier noch vom Benutzer erstellte Favoriten dargestellt werden. Diese könnten entweder nach Häufigkeit des Aufrufens erscheinen oder der Benutzer erstellt diese selber.



**Skizze 2 Prozessliste**

- Alle Prozesse werden einzeln als Metro-Kachel dargestellt. Von hier aus kann direkt auf den Erstellungslink mit dem Plus eines Prozess geklickt werden um eine neue zugehörige Prozessinstanz hinzuzufügen. Um einen neuen Prozess zu erstellen, kann auf die unterste Plus-Kachel geklickt werden.



**Skizze 3 Detail-Ansicht**

- Die in der SharePoint Webseite beinhaltenden Informationen könnten in der neuen Oberfläche wie in der obigen Skizzen angezeigt werden.

## 2.3 Evaluation Windows Store Apps (Metro UI)

Das neue Betriebssystem Windows 8 von Microsoft mit der neuen Benutzeroberfläche ist für die Öffentlichkeit erst seit dem 26. Oktober 2012 erhältlich. Für Studenten ist jedoch eine Vorab-Version verfügbar.

Diese muss man sich ohnehin installieren, da andererseits die Entwicklung der Applikation erschwert oder gar unmöglich gewesen wäre.

Um sich eine besseres Bild und Verständnis über die neue Benutzeroberfläche zu verschaffen, habe ich mir alle mitgelieferten Applikationen des neuen Betriebssystems angeschaut und ausprobiert, mit dem Hintergedanken, wie man diese Erfahrungen, Erkenntnisse und neuen Elemente für diese Studienarbeit verwenden kann.

Unter anderen wurde diese Applikationen näher bewertet:

### 2.3.1 Windows Store

Der Windows Store bietet grosse, aussagekräftige Kacheln:

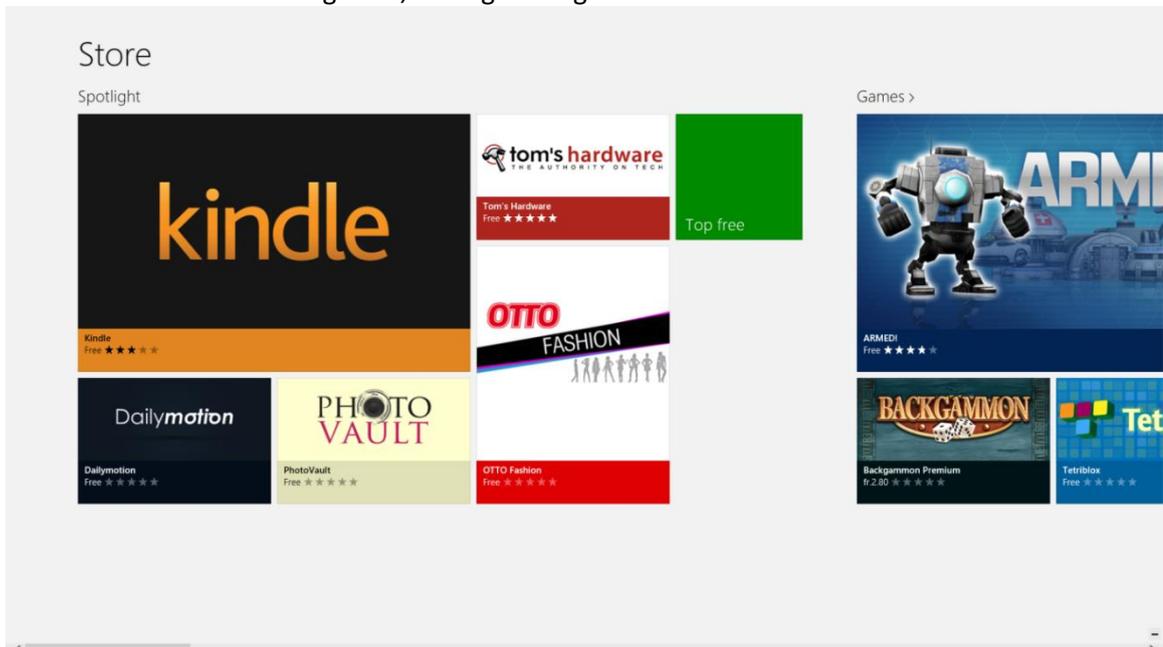


Abbildung 5 Windows Store

Vorteile:

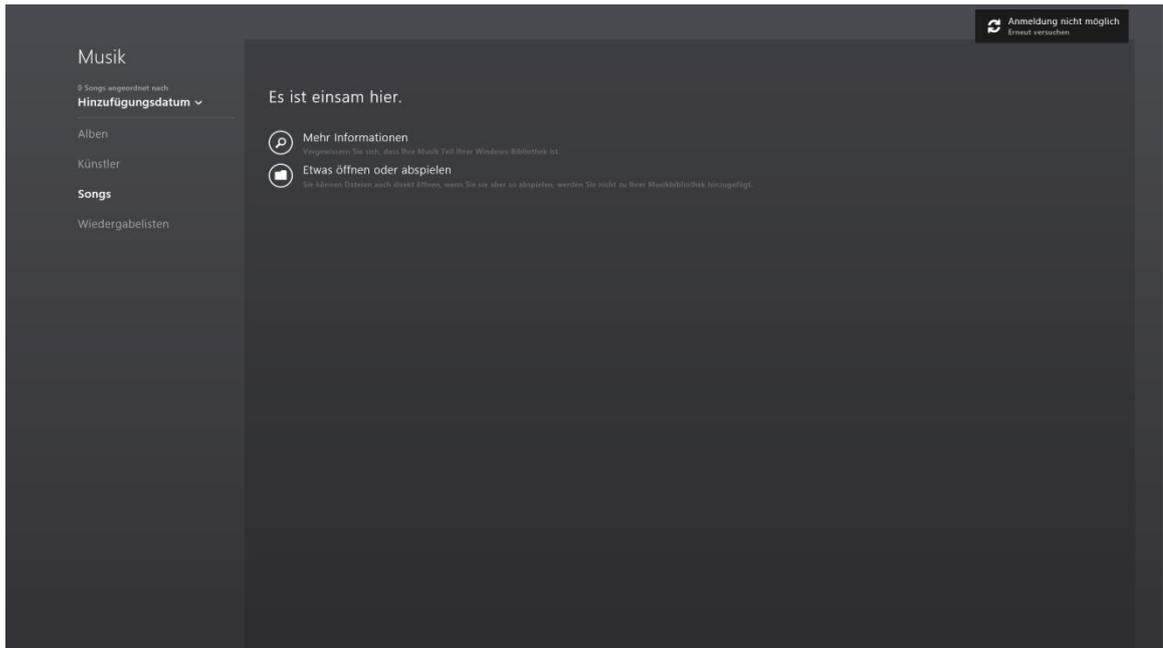
- Anpassbarkeit: Die Hauptfunktionalitäten können gross abgebildet und Favoriten können auf die Startseite gezogen werden was den schnellen Einstieg in die Applikation erlaubt.
- Grosse Bilder und Schriften: man braucht das Surface Tablet nicht näher an sich ziehen um den Inhalt lesen zu können.
- Fingerfreundlich

Nachteile:

- Schwierig, für BPMLight aussagekräftige Bilder zu finden.

### 2.3.2 Musik Applikation

Die Musik Applikation verwendet eher nüchterne Elemente:



**Abbildung 6 Windows Musik**

Vorteile:

- Navigation links sehr übersichtlich
- Layout rechts bietet viel Platz, was zu weniger horizontales scrollen bedeutet

Nachteil:

- Nichts aufregendes

### 2.3.3 Mockups

Nach den oben gesammelten Ideen und Einblicke in die neue Design-Sprache wurden Mockups erstellt.

In diesem Stadium konnten noch keine Möglichkeiten oder Einschränkungen des Frameworks miteinbezogen werden. Die Mockups dienen nur der besseren Visualisierung der Vorstellungen des Entwicklers.

Im Dokument Mockups.docx, aber auch auf Seite 13. 51, werden weitere Varianten vorgestellt.

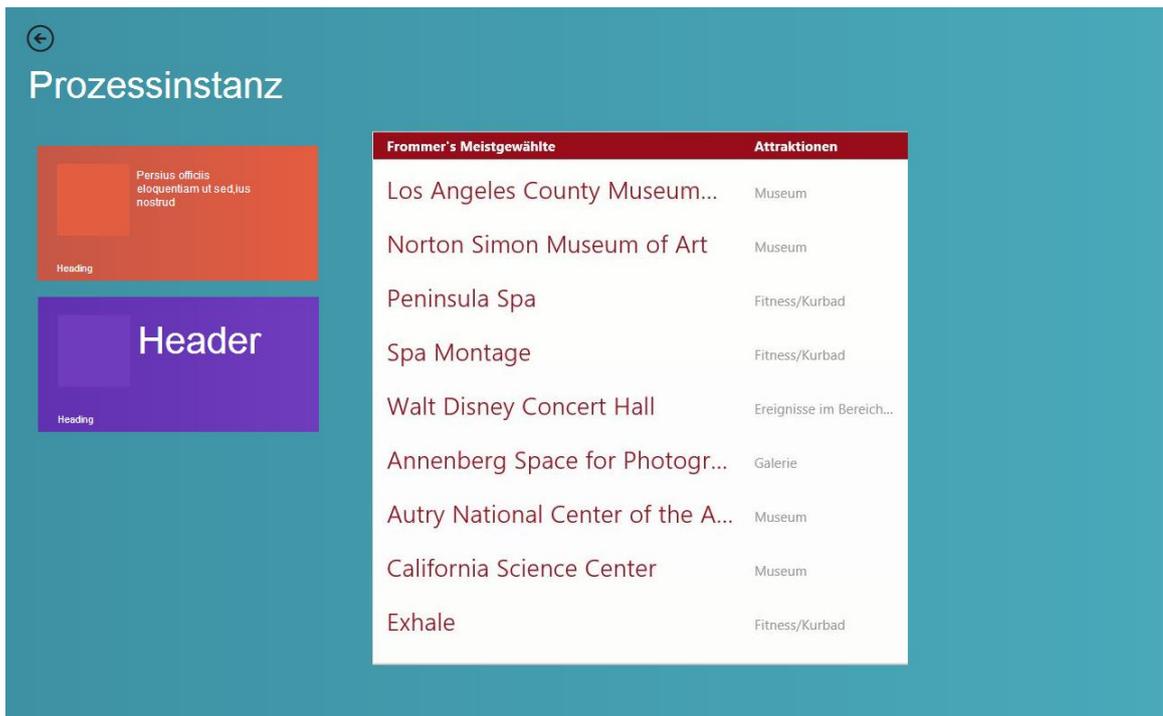
### 2.3.3.1 Startseite



**Abbildung 7 Mockup Startseite**

- Der grösste Platz wird von den Funktionen eingenommen, welche in der SharePoint-Implementation vorhanden sind.
  - ‚Prozesse‘ verlinkt auf eine Liste von allen Prozessen
  - ‚Instanzen‘ verlinkt auf eine Liste von allen Prozessinstanzen
  - ‚Administration‘ verlinkt auf alle Pendenzen
- Grosse Kacheln und Icons
  - Führt zu guter Lesbarkeit was bei einer Tablet Applikation sehr wichtig ist
- Bietet kleines Mass an Individualisierbarkeit indem der Benutzer Prozesse oder Prozessinstanzen welche er verfolgen will, als Favoriten auf die Startseite anheften kann.
  - Durch horizontales scrollen können beliebig viele Favoriten angezeigt werden.
- Pendenzen werden unter „Aktuell“ angezeigt. Unter Pendenzen versteht man Checklisten, welche vom Controller quittiert werden müssen
- Icons sind Metro-konform

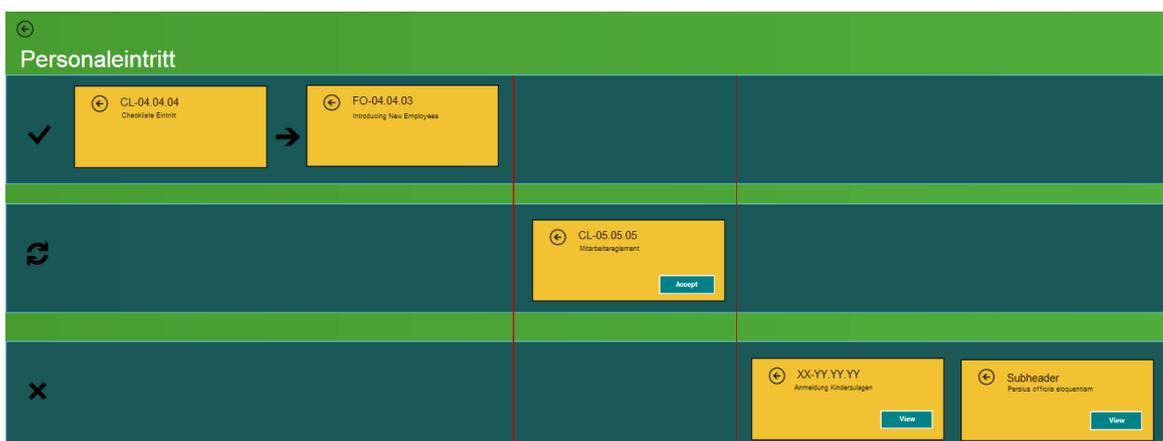
### 2.3.3.2 Listenansicht



**Abbildung 8 Mockup Listenansicht**

- Die Kacheln links können wiederum verschiedene Informationen anzeigen
  - Zum Beispiel: Instanz Name, Status, Avatar des zugewiesenen Benutzer usw.
- Als Alternative zu den Kacheln kann eine Liste mit allen Prozessinstanzen verwendet werden.

### 2.3.3.3 Detail-Ansicht



**Abbildung 9 Mockup Detail-Ansicht**

In dieser Ansicht werden die Checklisten einer einzigen Prozessinstanz angezeigt. Es gibt hier drei verschiedene Kategorien:

- Checklisten, welche ‚completed‘ und auch vom Controller quittiert worden sind
- Checklisten, welche auf die Quittierung durch den Controller warten
- Weitere Checklisten der Prozessinstanz, welche noch nicht bearbeitet worden sind.
- Die horizontalen Balken gruppieren die Kacheln und der vertikale rote Strich trennt sie.

## 2.4 Vorschlag endgültige Benutzeroberfläche

Nachdem etliche Versionen von den vorgestellten Mockups erstellt worden sind, hat man sich in Absprache mit dem Auftraggeber auf eine Benutzeroberfläche geeinigt. Diese ist eher schlicht und auf ein professionelles Auftreten getrimmt, und verzichtet deshalb auf verschieden bunte Kacheln.

Am 26. Oktober 2012 hat Microsoft einige Spezifikationen des Ziel-Tablets genannt. Das wirklich relevante für dieses Projekt ist die Auflösung des Bildschirms. Diese beträgt 1366 x 768 Bildpunkten. Aus Platzgründen in diesem Dokument kann aber nicht immer gewährleistet werden, dass bei den folgenden Screenshots das Seitenverhältnis immer stimmt.

### 2.4.1 Startseite

Startseite von BPM Light (Seitenverhältnis Microsoft Surface RT Tablet, 1366x768px, 16:9)

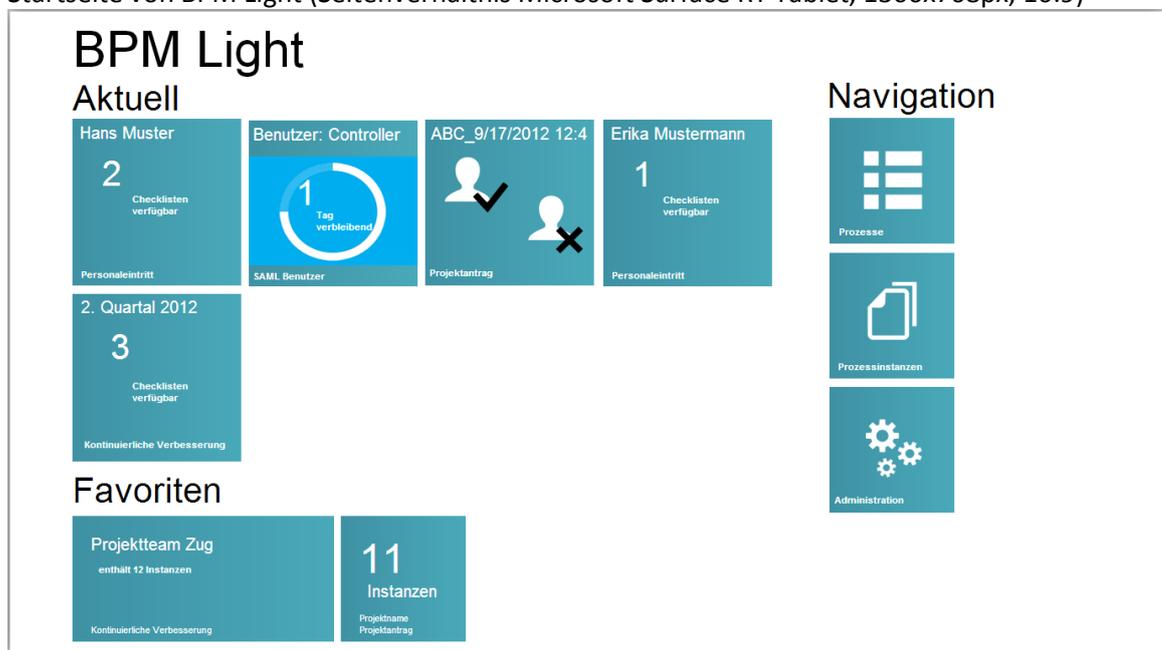


Abbildung 10 Startseite

#### 2.4.1.1 Aufbau

Der Aufbau lässt sich grundsätzlich in drei Gruppen unterteilen. Aktuell, Favoriten und Navigation.

##### Aktuell

Der Fokus des Benutzers sollte hierauf geleitet werden.

Unter Aktuell können drei verschiedene Typen von Kacheln angezeigt, die je nach Verwendungszweck verwendet wird. Die Bedeutung der einzelnen Kacheln und die Darstellung wird folgend dargestellt:

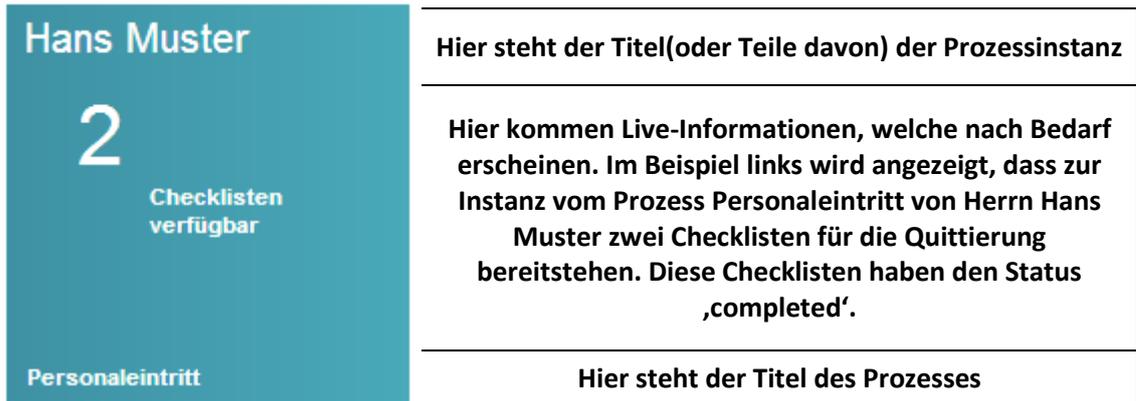


Abbildung 11 normale Aktuell-Kachel

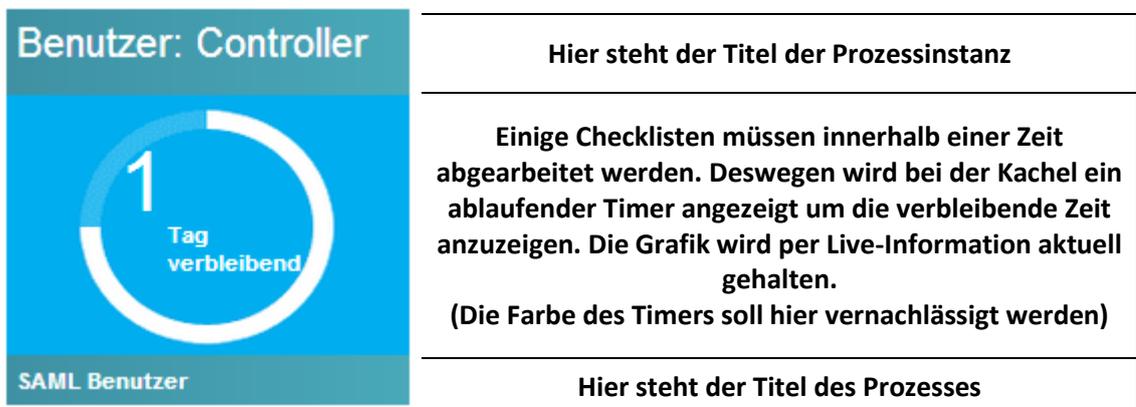


Abbildung 12 Timer

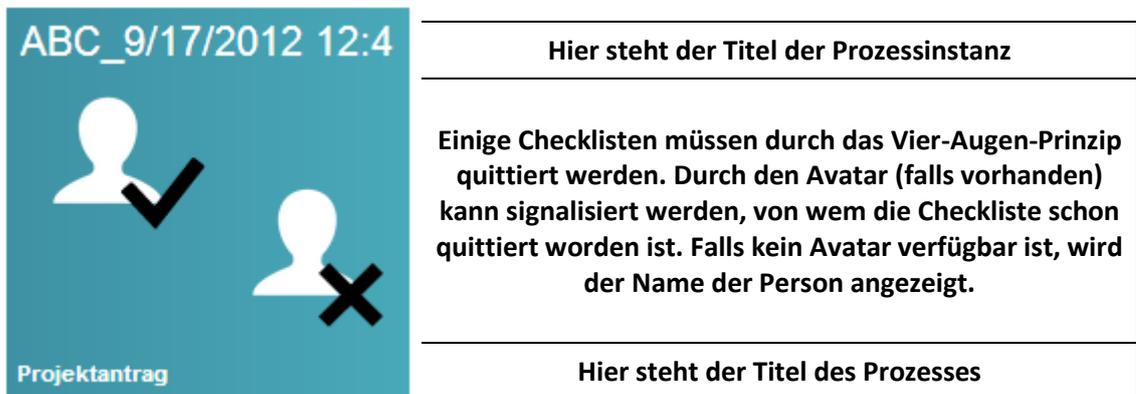


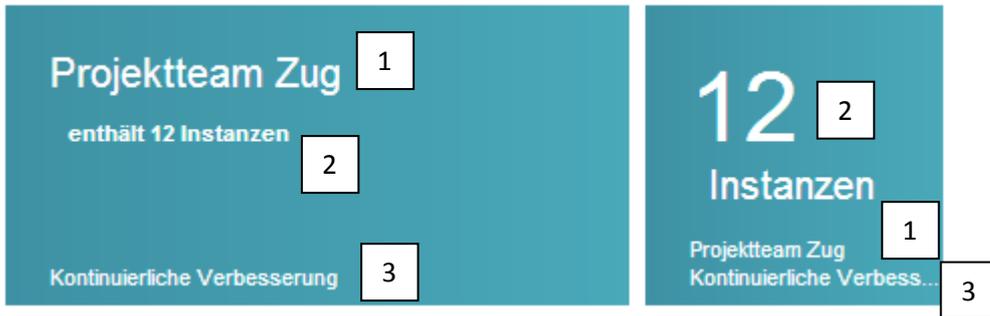
Abbildung 13 Vier-Augen-Prinzip

## Favoriten

Favoriten werden vom Benutzer erstellt. Er kann hier eine Menge von Prozessinstanzen eines Prozesses auswählen, welche er des Öfteren beobachten will. Bei Betrachtung einer Instanz aus einem Favoriten wird eine Ampel-Darstellung (wird weiter unten gezeigt) verwendet um den Status der beinhaltenden Checklisten anzuzeigen.

Diese Kacheln können in zwei verschiedenen Grössen angezeigt werden (310x150px und 150x150px, standardmässige Grösse von Metro Kacheln), wobei sie den gleichen

Informationsgehalt bieten können. Der Benutzer bekommt die Möglichkeit, den Titel selber zu definieren (im Beispiel: ‚Projektteam Zug‘).



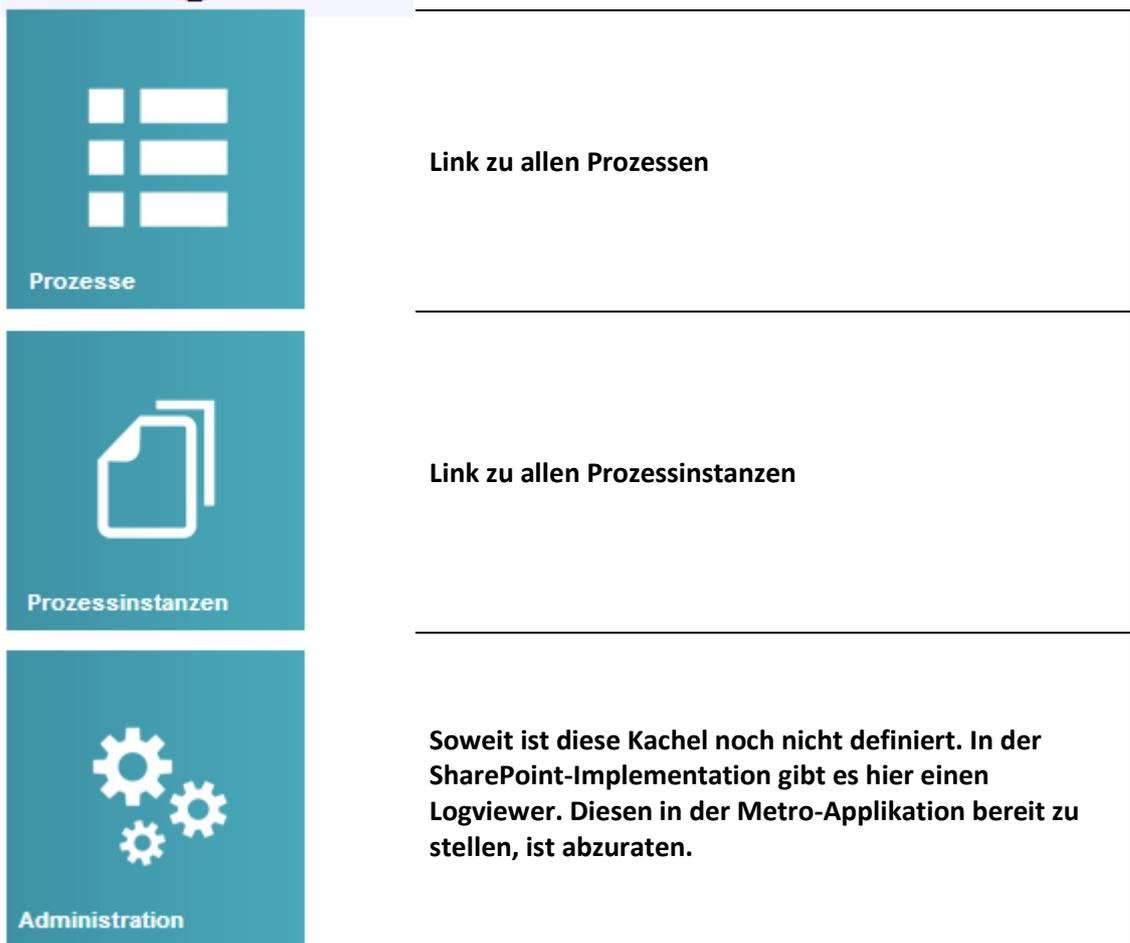
**Abbildung 14 Grosse und kleine Favoriten-Kachel**

Erklärung des Texts:

1. Vom Benutzer ausgewählter Name des Favoriten
2. Live-Information mit der Anzahl der beinhaltenden Prozessinstanzen
3. Name des Prozesses

## Navigation

# Navigation



**Abbildung 15 Navigations-Kacheln**

## 2.4.2 Prozesse

Jeder Prozess wird als eine eigenständige Kachel dargestellt und nicht als eine vertikal scrollbare Liste, da die Kacheln besser zum ganzen Gefüge passen und den Platz des Bildschirms verwerten.

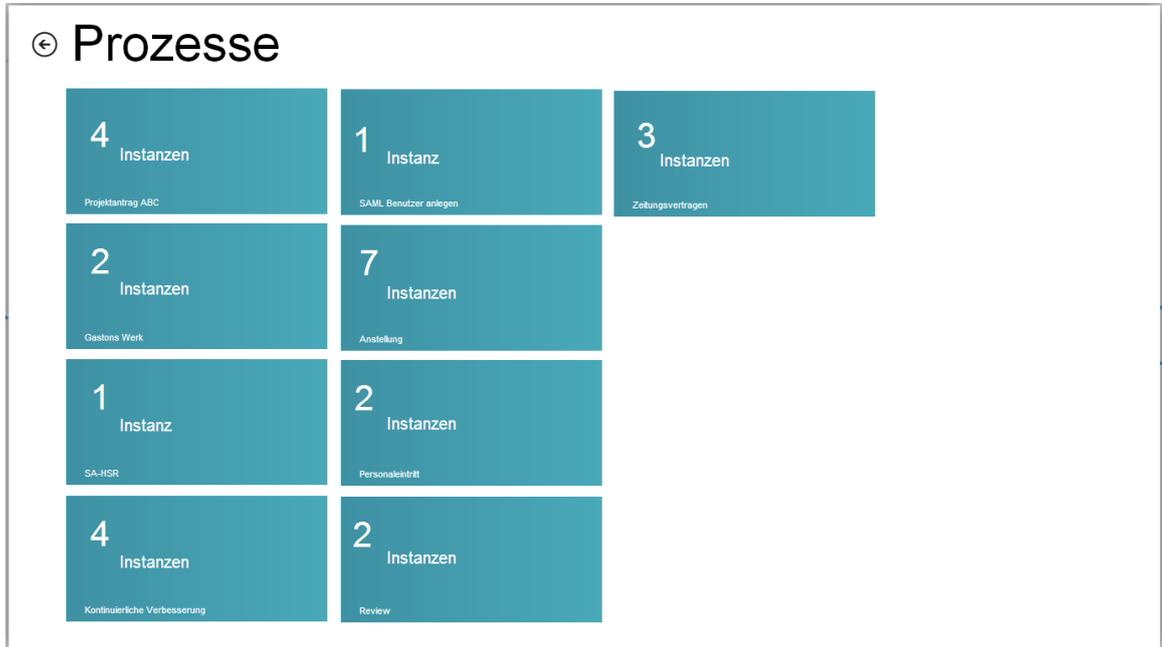


Abbildung 16 Alle Prozesse

Detail-Beschreibung einer Kachel:

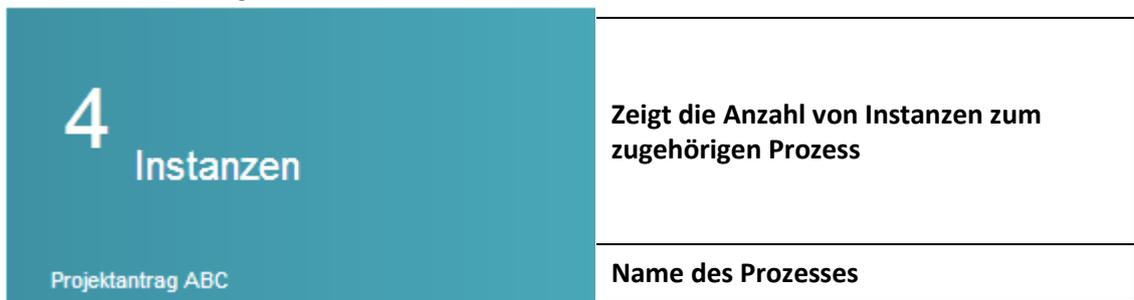


Abbildung 17 Grossansicht Prozess-Kachel

Jede dieser Kacheln ist anklickbar und führt zur Detailansicht welche weiter unten abgebildet ist.

### 2.4.3 Prozessinstanzen

Hier wird die gleiche Idee verfolgt, wie bei der Darstellung von den Prozessen.

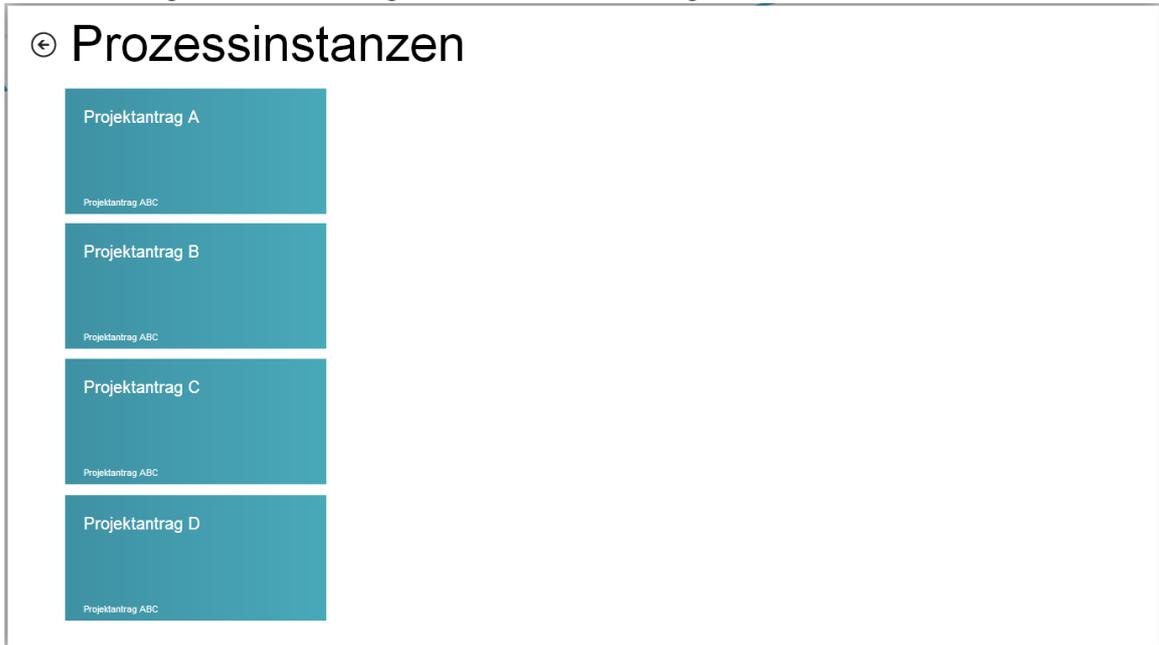


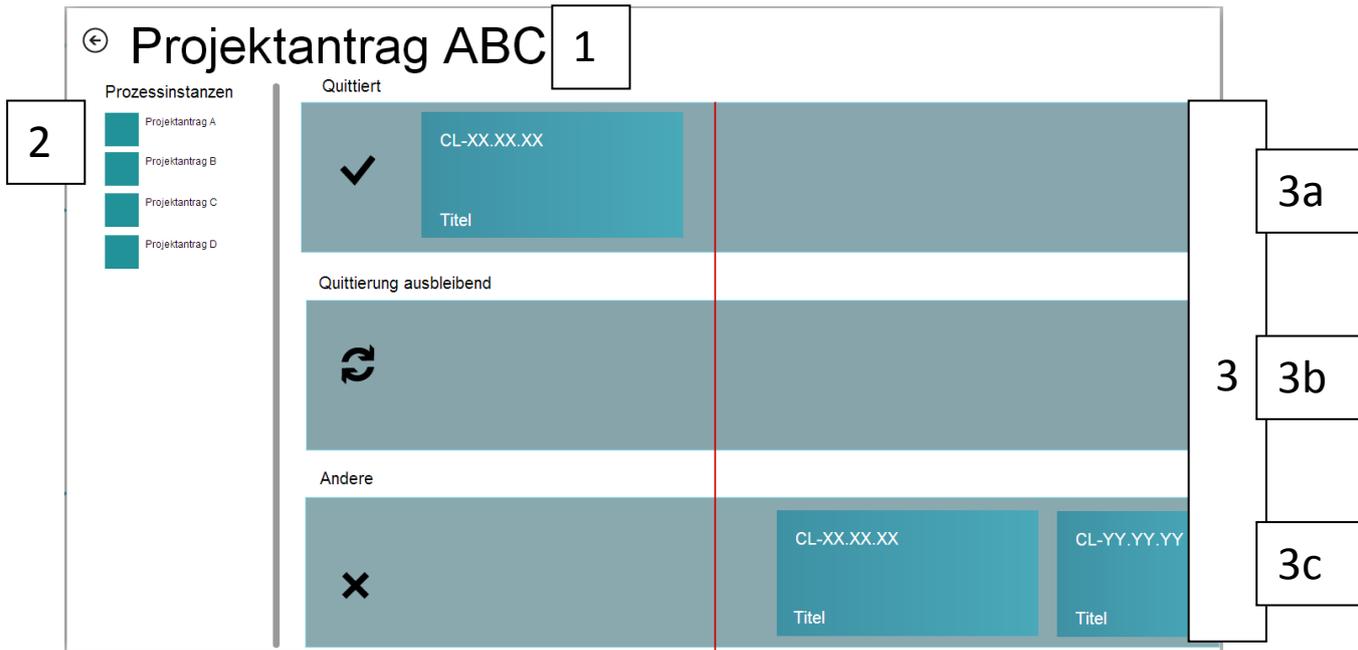
Abbildung 18 Alle Prozessinstanzen



Abbildung 19 Grossansicht Prozessinstanz-Kachel

### 2.4.4 Detailansicht

Um wiederverwertbare Elemente zu erstellen, bietet sich bei der Detailansicht eine Master/Detail-View zu verwenden, an. Darin können sowohl die Liste der Prozessinstanzen als auch die Zustände der Checklisten der zugehörigen Prozessinstanz angezeigt werden.

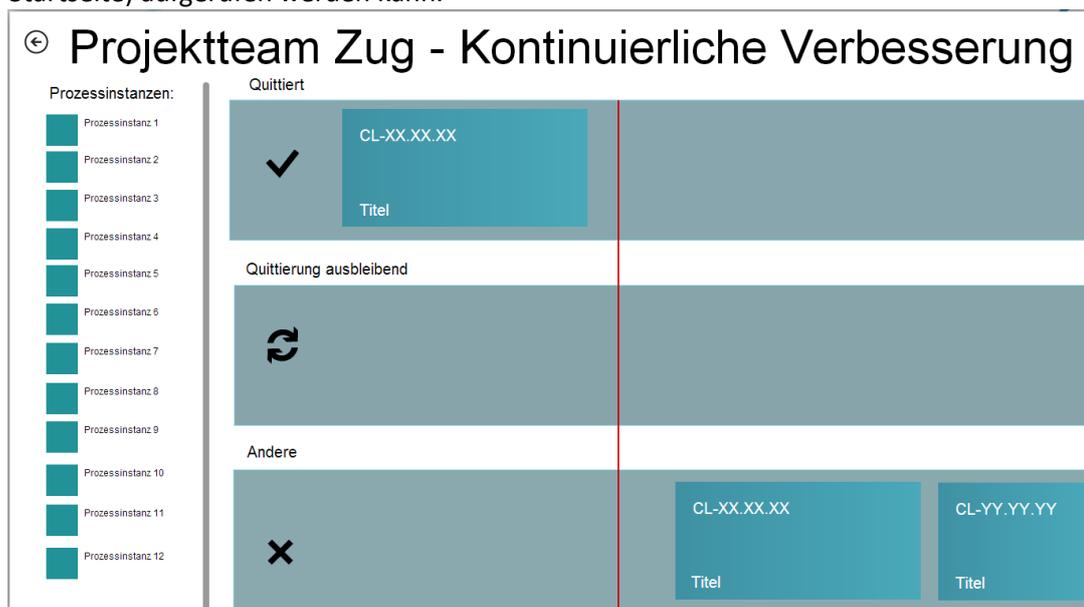


**Abbildung 20 Master/Detail-Ansicht**

1. Name des Prozesses
2. Liste der Prozessinstanzen
3. Darstellung der Checklisten einer ausgewählten Prozessinstanz (welche Prozessinstanz links ausgewählt ist, kann entweder durch farbliches hervorheben oder durch das Hinzufügen des Namen der Instanz im Titel [Nr.1 im Bild] angezeigt werden.)
  - a. Quittierte Checklisten
  - b. Zur Quittierung bereitstehenden Checklisten
  - c. Weitere Checklisten

### 2.4.5 Favoriten

Die Darstellung eines Favoriten bedient sich an der Master/Detail-Ansicht Konzept. Folgender Mockup wird hier gezeigt, da der Favorit direkt aus der Startseite (Abbildung 10 Startseite) aufgerufen werden kann.



**Abbildung 21 Favoriten Darstellung**

## 2.4.6 Checkliste quittieren

Als nächstes wird beschrieben, wie eine Quittierung abläuft wenn man diese direkt aus der Startseite aufgreift. Das Prinzip wird auch bei der Master/Detail-Ansicht verwendet.

1. Der Benutzer sieht, dass im mittleren horizontalen Balken eine Checkliste zum Quittieren bereit steht:

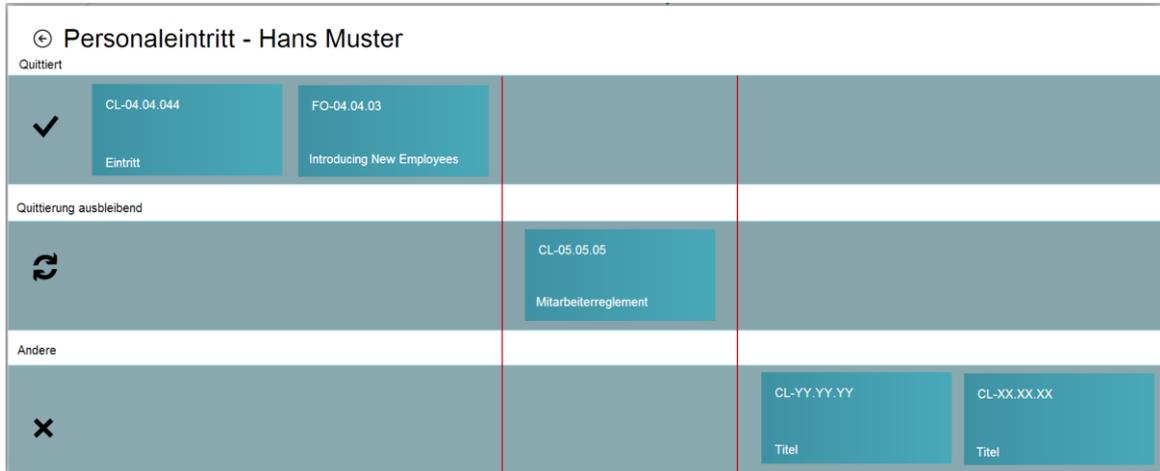


Abbildung 22 Checkliste quittieren

2. Der Benutzer klickt die entsprechende Kachel. Es erscheint dieser Dialog:

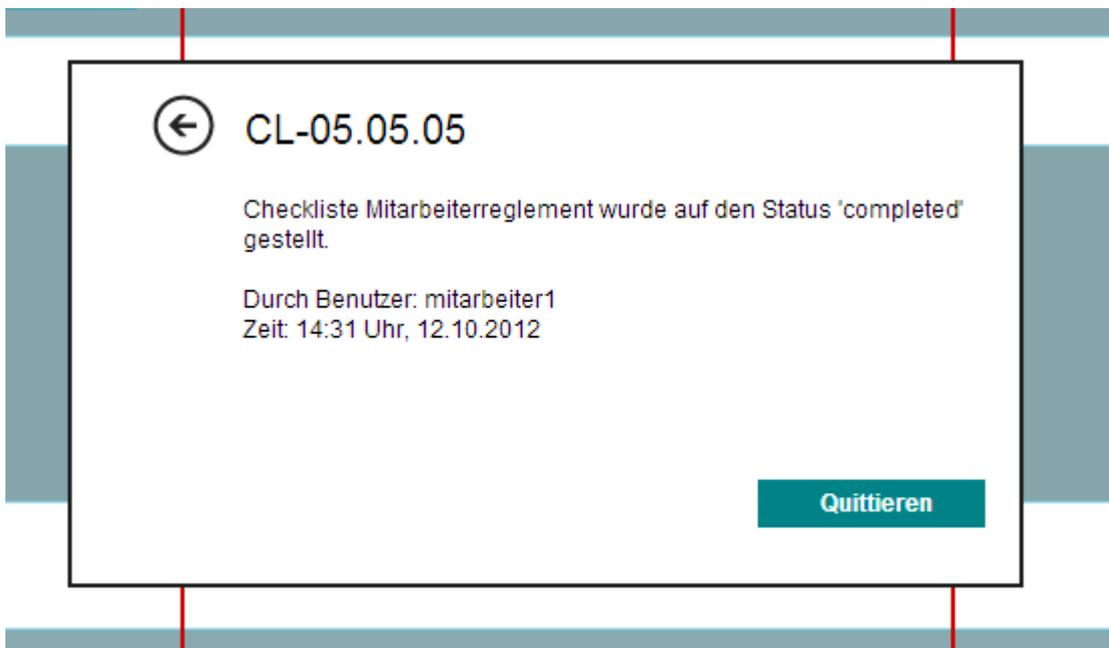


Abbildung 23 Quittierungs-Bestätigung

3. Der Benutzer quittiert die Checkliste.  
Der Dialog kann dazu benutzt werden, um anzuzeigen, durch welchen Benutzer und wann diese Checkliste auf den Status ‚completed‘ gestellt wurde.  
Diese quittiert Checkliste, bzw. die Kachel die sie repräsentiert, rutscht nun zum oberen Balken zu den anderen quittierten Checklisten.

Personaleintritt - Hans Muster

Quittiert

✓	CL-04.04.044 Eintritt	FO-04.04.03 Introducing New Employees	CL-05.05.05 Mitarbeiterreglement
---	--------------------------	--	-------------------------------------

Quittierung ausbleibend

Andere

✗	CL-YY.YY.YY Titel	CL-XX.XX.XX Titel
---	----------------------	----------------------

Abbildung 24 Checkliste quittiert

## 2.5 Prototyp

Die erstellten Mockups wurden versucht, so ähnlich wie möglich im Prototyp widerzuspiegeln.

### 2.5.1 Einschränkungen

Während der Einarbeitung in die neue Technologie von Windows Store Apps wurde mit der Zeit klar, dass die Technologie einige Einschränkungen mit sich bringt. Da das neue Betriebssystem Windows 8 (oder Varianten davon, siehe Windows RT) auf verschiedenartige Geräte-Typen (Desktop-PCs, Tablets und Smartphones) installiert sein wird, die sehr unterschiedliche Hardwarekonfigurationen mit sich bringen können, müssen Einschränkungen in der Technologie gemacht werden.

Es gibt zum Beispiel keine offiziell unterstützte Möglichkeit, um eine lokale Datenbank in einer Windows Store App zu verwenden, da der Namespace `System.Data` fehlt [1]. Stattdessen sollen die Informationen der Applikation über einen Web Service über das Internet bezogen werden. Mit diesem Lösungsansatz können die oben beschriebenen verschiedenen Hard- und Softwarekonfigurationen als Fehlerquelle ausgeschlossen werden. Andererseits wird stets eine bestehende Internet-Verbindung vorausgesetzt.

Bei näherer Betrachtung des Web Services, welcher durch den Auftraggeber zur Verfügung gestellt wurde und beim Versuch, diesen mit der Windows Store App zu konsumieren, wurde die nächste Einschränkung aufgedeckt. Nach aktuellem Stand (Mitte Oktober 2012) können Windows Store Applikation nicht mit einem Web Service vom Typen SOAP v1.2 umgehen. Es kann lediglich der Typ SOAP v1.1 eingesetzt werden. [2]

### 2.5.2 Erweiterungen

Da in der SharePoint Webseite die Idee von Aktuell-Objekten, welche zu quittierende Pendenzen anzeigen, nicht vorgesehen war, muss die bestehende Infrastruktur von BPMLight erweitert werden.

Da eine Aktuell-Kachel im Endeffekt eine Checkliste repräsentiert müssen diese als Eintrag in der Datenbank vorhanden sein.

Somit ist es nötig, eine neue Tabelle mit dem Namen ChecklistEntry in die Datenbank einzufügen:

**Tabelle 2 ChecklistEntry**

checklistId	checklistName	checklistStateId	instanceId	instanceStateId
-------------	---------------	------------------	------------	-----------------

Das Feld checklistStateId referenziert einen Eintrag aus der Tabelle ChecklistState:

**Tabelle 3 ChecklistState**

id	description
0	accepted
1	pending
2	other

### 2.5.3 Entwicklungsumgebung

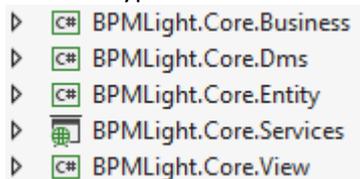
Für die Entwicklung des Prototyps wurden folgende Soft- und Hardware verwendet:

- Microsoft Windows 8 (Pro)
  - Mindestvoraussetzung für Entwicklung von Windows Store Apps
- Microsoft Visual Studio 2012 (Ultimate)
  - Mindestvoraussetzung für Entwicklung von Windows Store Apps
  - Verwendung von C# für Programmcode und XAML für die Benutzeroberfläche
- Microsoft Visual Studio 2010 Ultimate
  - Wurde eingesetzt, um sich in die bestehende Lösung von BPMLight, der SharePoint-Webseite, einzuarbeiten.
- Microsoft SQL Server 2012
  - Datenbank mit Beispiel-Einträgen der SharePoint Webseite
- soapUI 4.5.1 64bit
  - Ein Tool, um einen Web-Service lokal zu nachzuahmen (Mocking).
- Samsung Series 7 Slate PC XE700T
  - Windows 8 Tablet mit gleicher Bildschirmauflösung (1336x768 Bildpunkte) wie das Ziel-Tablet

### 2.5.4 Wiederverwendeter Code

Um sich den Aufwand so sehr wie möglich zu verkleinern, wurde so viel Code wie nötig und so wenig wie möglich aus der bestehenden Lösung von BPMLight übernommen um einen lauffähigen Prototypen zu erstellen.

Im Prototyp wurden schliesslich folgende Teilprojekte verwendet.

**Abbildung 25 Wiederverwendeter Code**

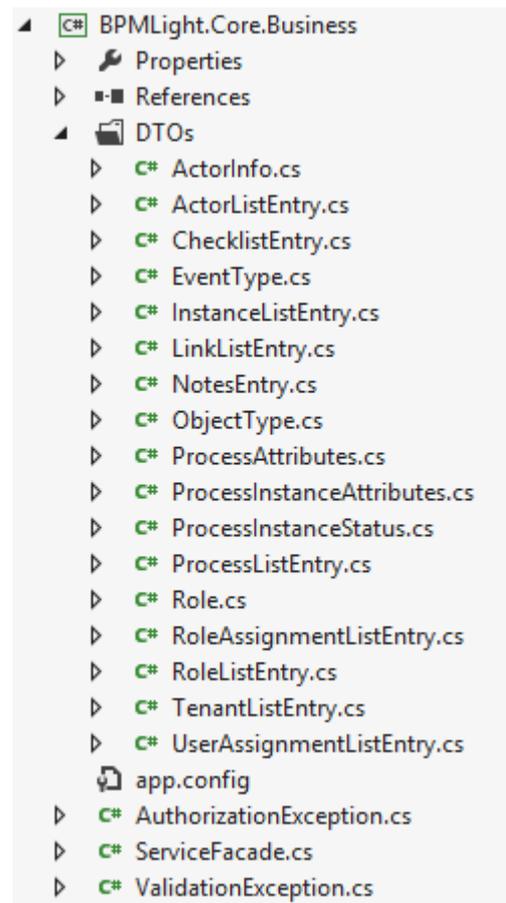
Mit Ausnahmen der im nächsten Unterkapitel beschriebenen Änderungen, sind keine anderen Änderungen vorgenommen worden.

### 2.5.4.1 BPMLight.Core.Business

Hierbei handelt es sich um die Business-Logik und die Definition der zu übermittelten Data Transfer Objects (DTO).

Als einzige Ergänzung in dieser Lösung ist die Datei ChecklistEntry.cs aufzuführen.

Die dafür definierten `DataMember` können von der Tabelle 2 ChecklistEntry entnommen werden.



### 2.5.5 Mock Web Service

Um den Prototyp zu erstellen, musste ein konformer Web Service vom Typ SOAP v1.1 erstellt werden. Durch Wiederverwendung der bestehenden Lösung konnte dies schnell bewerkstelligt werden.

Die .wsdl-Datei befindet sich unter `source\BPMLight-NewUI\BPMLight-NewUI\ServiceReferences\ServiceReference`.

Die folgenden Methoden wurden im Prototyp verwendet und wurden komplett übernommen:

```

/// <summary>
/// Lists all processes.
/// </summary>
/// Replaces GetAvailableProcesses(int tenantId)
/// <param name="actor">Actor information for authorization, logging, ...</param>
/// <returns>A list of all processes</returns>
[OperationContract]
List<ProcessListEntry> ListProcesses(ActorInfo actor);

/// <summary>
/// Lists all instances.
/// </summary>
/// <param name="actor">Actor information for authorization, logging, ...</param>
/// <returns>A list of all instances</returns>
[OperationContract]
List<InstanceListEntry> ListInstances(ActorInfo actor);

/// <summary>
/// Lists all instances of a given process.
/// </summary>
/// Replaces GetAllProcessInstancesForProcess(int processId)
/// <param name="actor">Actor information for authorization, logging, ...</param>
/// <param name="processId">The process id</param>
/// <returns>A list of all instances of a given process</returns>
[OperationContract]
List<InstanceListEntry> ListInstancesByProcess(ActorInfo actor, int processId);

```

### Abbildung 26 Übernommene Methoden

Der Web Service musste um folgende Methoden erweitert werden:

```

/// <summary>
/// Lists all checklists of a given process.
/// </summary>
/// <param name="actor">Actor information for authorization, logging, ...</param>
/// <param name="instanceId">The instance id of the process</param>
/// <returns>A list of all checklists of a given process instance</returns>
[OperationContract]
List<ChecklistEntry> ListChecklistsByProcessInstance(ActorInfo actor, int instanceId);

/// <summary>
/// Lists all checklists of a given status
/// </summary>
/// <param name="actor">Actor information for authorization, logging, ...</param>
/// <param name="statusId">The status id</param>
/// <returns>A list of all checklists of a given status</returns>
[OperationContract]
List<ChecklistEntry> ListChecklistByStatus(ActorInfo actor, int statusId);

/// <summary>
/// Lists a single instance of a given instanceId
/// </summary>
/// <param name="actor">Actor information for authorization, logging, ...</param>
/// <param name="instanceId">The instance id</param>
/// <returns>A single InstanceListEntry of a given instanceId</returns>
[OperationContract]
List<InstanceListEntry> ListSingleInstanceById(ActorInfo actor, int instanceId);

```

### Abbildung 27 Hinzugefügte Methoden

### 2.5.5.1 soapUI

Für das Mocking des Web Service wurde das Programm soapUI von SmartBear verwendet. [3]

Dieses Programm ist während der Entwicklungszeit notwendig um die Applikation zu testen.

Nach der Installation des Programmes kann eine .xml-Datei importiert werden, welche die Antworten des Web Services gespeichert hat.

Diese .xml-Datei ist unter dem Ordner ..\soapUI\BPMLight-soapui-project.xml abgespeichert.

### 2.5.6 Windows Store App

Im diesem Unterkapitel wird die erarbeitete Lösung des Prototypen näher beschrieben

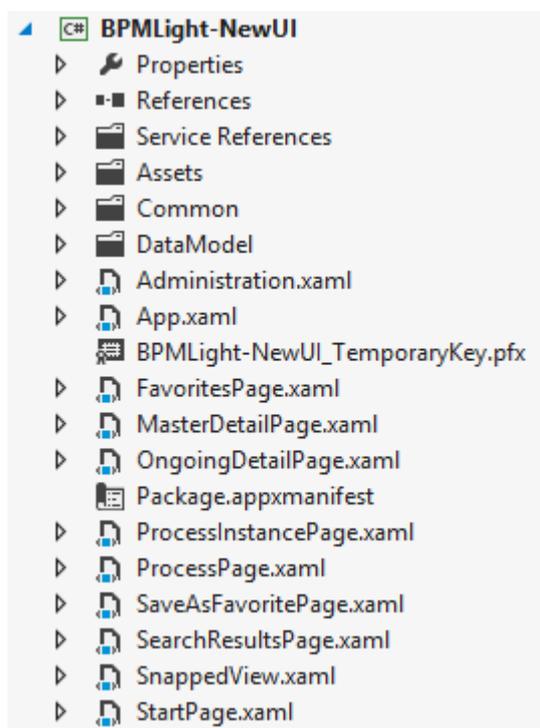


Abbildung 28 Struktur Windows Store App

#### 2.5.6.1 Common

Der Common Ordner beinhaltet einige nützliche Dateien. Diese erleichtern die Entwicklung von Windows Store Apps, z.B. die verschiedenen Style-Vorlagen in StandardStyles.xaml, oder werden vorausgesetzt, wie LayoutAwarePage.cs.

Das Aussehen einer typischen Applikations-Kachel wird in StandardStyles.xaml definiert. Da sich dort eine grosse Menge von verschiedenen Styles befinden, wurden die für den Prototyp

notwendigen Definitionen für die Kacheln in der Datei CustomStyles.xaml abgelegt.

Hier ein Beispiel einer Definition:

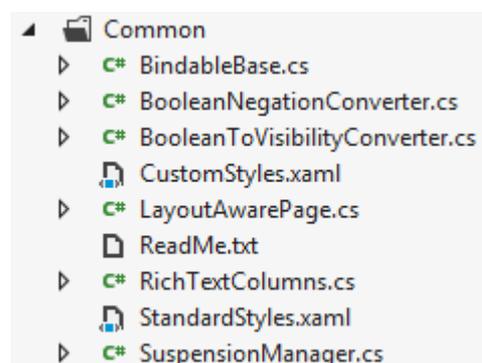


Abbildung 29 Common Ordner

```

<DataTemplate x:Key="FavoritenItemTemplate">
  <Grid HorizontalAlignment="Left" Width="180" Height="180">
    <Border Background="{StaticResource BlockBackgroundBrush}" />
    <TextBlock Text="{Binding GroupName}"
      Style="{StaticResource GridTitleTextStyle}" Foreground="White" />
    <TextBlock Text="{Binding NrOfEntries}"
      Style="{StaticResource GridDescriptionTextStyle}" Foreground="White" Fo
    <TextBlock Text="Instances"
      Style="{StaticResource GridDescriptionTextStyle}" Foreground="White"
      Margin="35,0,0,55"/>
    <StackPanel VerticalAlignment="Bottom" Orientation="Horizontal"
      Background="{StaticResource ListViewItemOverlayBackgroundThemeBrush}">
      <TextBlock Text="{Binding ProcessName}" Margin="12,4,0,8" Foreground=
        Height="42" />
    </StackPanel>
  </Grid>
</DataTemplate>

```

**Abbildung 30** Definition ItemTemplate

Diese Definition entspricht Abbildung 14 auf Seite 12. Hier nochmals die gleiche Abbildung:



**Abbildung 31** Favoriten Kachel

Mithilfe des WPF DataBindings werden in der obigen Definition viele Informationen auf der Kachel dargestellt.

Der `<Border>` -Tag zum Beispiel referenziert auf die Ressource, welche die Farbe der Kachel definiert:

```

<LinearGradientBrush x:Key="BlockBackgroundBrush" EndPoint="0.5,1" StartPoint="0.5,0">
  <GradientStop Color="#FF37AFF9" Offset="1"/>
  <GradientStop Color="#FF37AFF9" Offset="0"/>
</LinearGradientBrush>

```

**Abbildung 32** Farbdefinition Kachel

Die weiteren `<TextBlock>` -Tags referenzieren den Gruppennamen, die Anzahl enthaltener Instanzen und den Prozessnamen der Favoriten-Kachel.

## 2.5.6.2 DataModel

### DataSource

Für einige Seiten der Applikation muss hier eine Helfer-Klasse implementiert werden. Diese Klassen dienen dazu, die Verbindung mit dem Web Service aufzunehmen, SOAP Nachrichten

zu versenden und zu empfangen. Die Resultate davon werden jeweils in einer `ObservableCollection<>` des jeweiligen Typ gespeichert.

## Items

Hier werden die Objekte definiert, die auf der StartPage repräsentiert werden.

## Asynchrone Programmierung

Um immer eine flüssige Bedienung der Benutzeroberfläche zu gewährleisten, werden die Objekte für die DataSource und Items asynchron programmiert.

Durch Kennzeichen `async` einer Methode, wird es möglich, den `await` Operator zu verwenden. Im folgenden Beispiel ist diese Technik sehr hilfreich, wenn die Antworten des Web Service aufgrund der manchmal Unberechenbarkeit der Internet-Qualität unterschiedlich lang dauern können:

```
private async Task<IEnumerable<InstanceListEntry>> GetInstancesByProcessIdAsync
                                                    (int processId)
{
    try
    {
        IService service = new ServiceClient(new BasicHttpBinding(),
            new EndpointAddress(
                new Uri("[..]mockBasicHttpBinding_IService1?WSDL")));

        this.masterItems = await service.ListInstancesByProcessAsync(
            new ActorInfo(), processId);

        return this.masterItems;
    }
    catch (Exception e)
    {
        throw new NotImplementedException();
    }
}
```

Abbildung 33 Asynchrone Methode

### 2.5.6.3 StartPage

Auf der StartPage werden so viele Informationen wie mögliche auf einen Blick angezeigt, ohne dass horizontal oder vertikal gescrollt werden muss.

Dabei lässt sich diese Seite in drei Raster (Grids) aufteilen. Hier mit rotem Rand eingezeichnet:

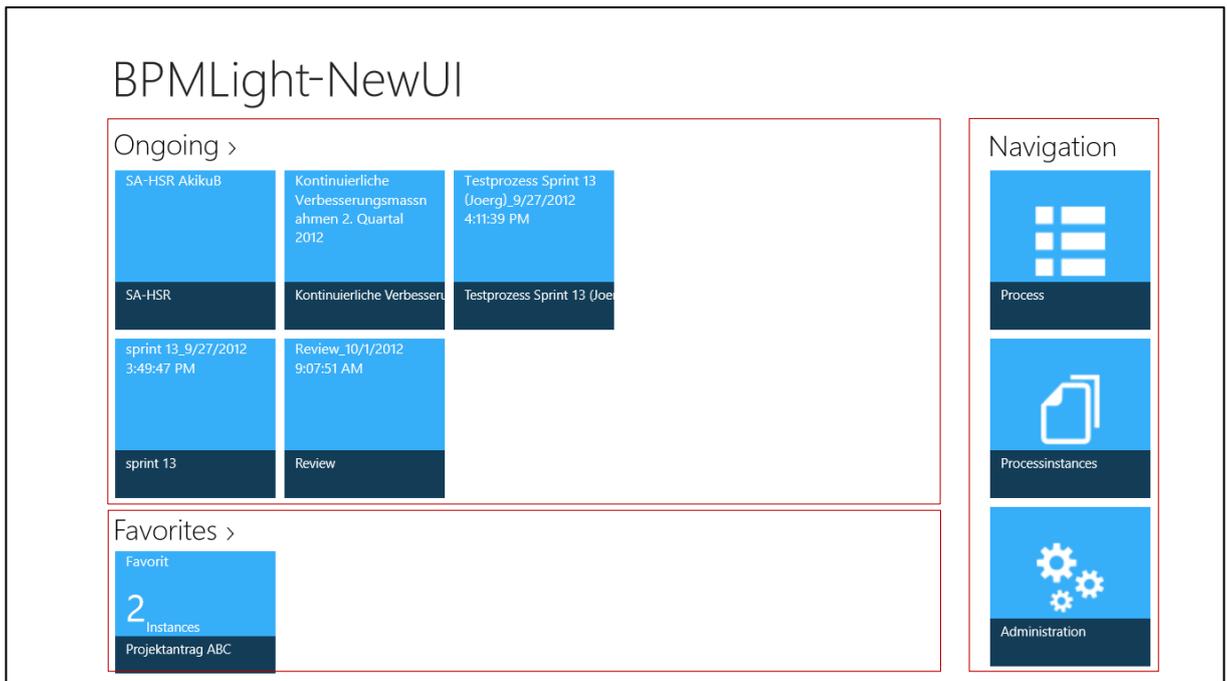


Abbildung 34 Startseite mit Raster

Da die Bildschirmfläche begrenzt ist, wird auf der StartPage nur eine begrenzte Anzahl von Kacheln angezeigt. Im Raster „Ongoing“ können aus diesem Grund maximal 10 solche Kacheln und im Raster „Favorites“ nur 5 davon angezeigt werden. Falls jedoch mehr Informationen zu zeigen sind, kann jeweils auf eine Kopfzeile mit kleinem Pfeil geklickt werden. Dies führt zu einer Ansicht mit allen verfügbaren Elementen. Siehe hier im Beispiel von „Ongoing“:

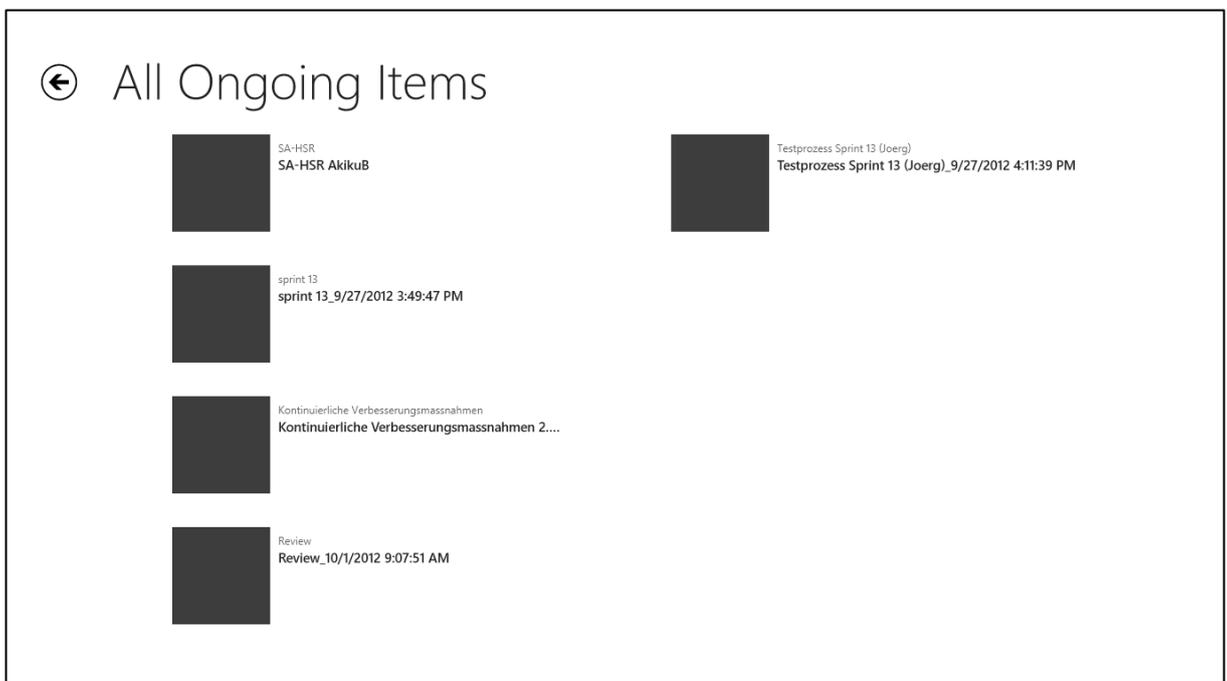


Abbildung 35 Detail-Ansicht

Entsprechend gibt es die gleiche Ansicht für die Elemente von „Favorites“.

Für die drei verschiedenen Raster muss folgender Code geschrieben werden:

1. Zuerst muss man als `<CollectionViewSource>` angeben, wie die jeweilige Quelle der Daten heisst:

```
<CollectionViewSource x:Name="aktuellItemsViewSource" Source="{Binding Aktuell}"/>
<CollectionViewSource x:Name="favoritenItemsViewSource" Source="{Binding Favoriten}"/>
<CollectionViewSource x:Name="navigationItemsViewSource" Source="{Binding Navigation}"/>
```

2. Die oben genannten Quellen müssen im C#-Code im `DefaultViewModel` mit Daten gefüllt werden:

```
var topOngoingDataGroup = OngoingDataItem.TopOngoingItems;
this.DefaultViewModel["Aktuell"] = topOngoingDataGroup;

var favoritenDataGroup = FavoriteDataItem.TopFavoriteDataItems;
this.DefaultViewModel["Favoriten"] = favoritenDataGroup;

var navigationDataGroup = StartPageDataSource.GetNavigationDataGroup();
this.DefaultViewModel["Navigation"] = navigationDataGroup;
```

3. Die im ersten Schritt definierten Quellen können nun dem jeweiligen Raster zugewiesen werden indem man sie als `ItemsSource=""` angibt:

```
<GridView
    ScrollViewer.HorizontalScrollMode="Disabled"
    x:Name="aktuellGridView"
    IsSwipeEnabled="false"
    IsItemClickEnabled="True"
    SelectionMode="None"
    ItemClick="AktuellView_ItemClick"
    Grid.Row="1" Grid.Column="0"
    ItemsSource="{Binding Source={StaticResource aktuellItemsViewSource}}"
    ItemTemplate="{StaticResource AktuellItemTemplate}" Margin="118,40,5,0"/>

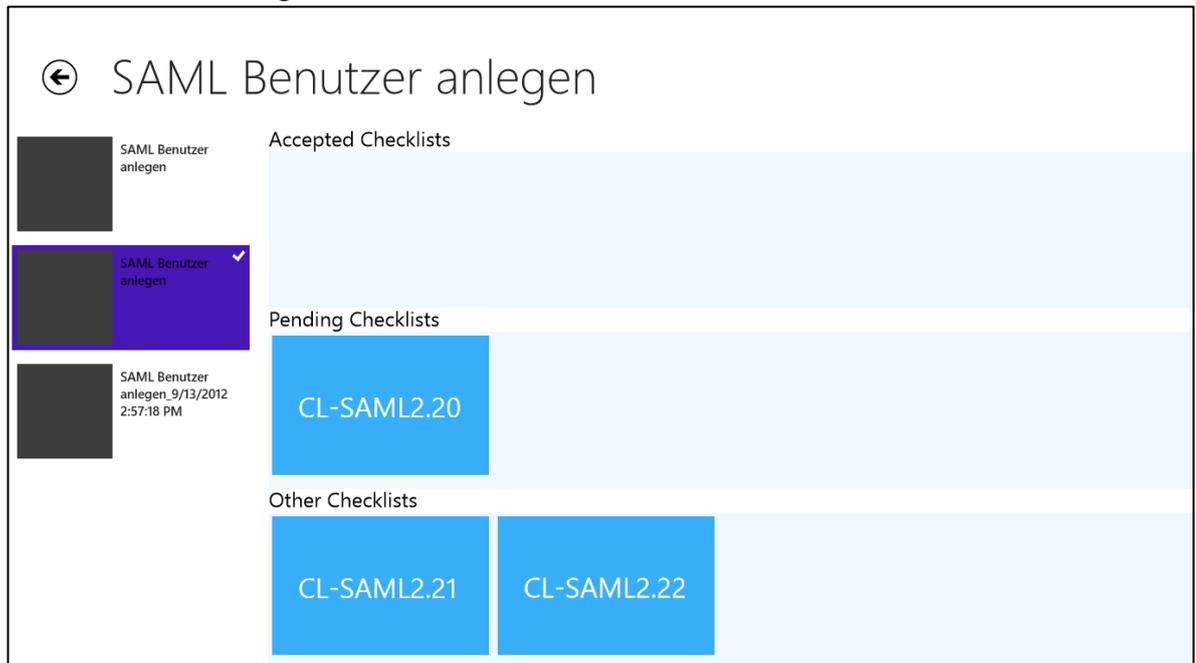
<GridView
    Grid.Column="0" Grid.Row="2" VerticalAlignment="Center" Margin="118,40,0,0"
    ScrollViewer.HorizontalScrollMode="Disabled"
    x:Name="favoritenGridView"
    AutomationProperties.AutomationId="favoritenGridView"
    AutomationProperties.Name="Grouped Items"
    SelectionMode="None"
    IsSwipeEnabled="false"
    IsItemClickEnabled="True"
    Width="Auto"
    ItemClick="FavoritenItem_ItemClick"
    ItemsSource="{Binding Source={StaticResource favoritenItemsViewSource}}"
    ItemTemplate="{StaticResource FavoritenItemTemplate}" />

<GridView
    ScrollViewer.HorizontalScrollMode="Disabled"
    x:Name="navigationGridView"
    AutomationProperties.AutomationId="navigationGridView"
    AutomationProperties.Name="Grouped Items"
    SelectionMode="None"
    IsSwipeEnabled="false"
    IsItemClickEnabled="True"
    ItemClick="NavigationItem_ItemClick"
    Grid.Row="1" Grid.Column="1" Grid.RowSpan="2"
    ItemsSource="{Binding Source={StaticResource navigationItemsViewSource}}"
    ItemTemplate="{StaticResource NavigationItemTemplate}" Margin="0,40,0,0"/>
```

- In jeder Definition eines Rasters wird auch das Aussehen der Kachel referenziert, indem man unter `ItemTemplate="{StaticResource XXX}"` die entsprechende Vorlage auswählt.

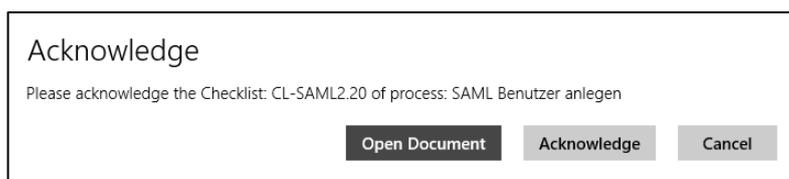
#### 2.5.6.4 MasterDetailPage

Während der Entwicklung wurde immer mehr klar, was genau für Elemente in dieser Ansicht verwendet werden konnten. Das Konzept einer wiederverwendbaren Ansicht mit einer Liste von Prozessinstanzen auf der linken Seite und die drei Gruppen von Checklisten auf der rechten Seite konnten gut übernommen werden



**Abbildung 36 Master-/Detail-Ansicht**

Bei Klick der zu quittierende Kachel sind folgende Optionen im Dialog gegeben:



**Abbildung 37 Quittierungs-Dialog**

Für die anderen Checklisten sind im Moment keine solchen Optionen vorgesehen. Die Funktionen ‚Open Document‘ und ‚Acknowledge‘ werden in dieser Studienarbeit auch nicht implementiert, da dafür weitreichende Änderungen an der Infrastruktur von BPMLight vorgenommen werden müssen.

Zum Beispiel hat man über den Web Service keinen Zugriff auf die Dokumente und wo die Dokumente liegen ist auch nicht bekannt. Zudem kann das quittieren mit dem Mock Web Service und soapUI auch nicht bewerkstelligt werden, da damit keine Datenmanipulation bewerkstelligt kann.

### 2.5.6.5 Favorites

In der Master-/Detail-Ansicht eines Prozesses können Favoriten erstellt werden. Dazu muss man mehrere Prozessinstanzen auswählen und es erscheint eine AppBar mit der Option, einen Favoriten zu speichern.

Bei Fingerbedienung muss dafür eine Cross-Slide Geste durchgeführt werden. Das heisst, man hält den Finger auf eine Prozessinstanz drauf und zieht dann diese horizontal weg.

Die AppBar kann auch durch eine Wischbewegung vom unteren Bildschirmrand aufgerufen werden.

Die Favoriten eines Benutzers werden lokal auf das Gerät gespeichert. Über eine Speicherung in die Microsoft Cloud wurde abgesehen, da das Verhalten davon mit einem Gerät nicht getestet werden konnte.

Die Löschung eines Favoriten wird ebenfalls in der Master-/Detail-Ansicht durchgeführt indem man auf den entsprechenden Befehl in der AppBar klickt. Dabei wird der ganze Favorit gelöscht.

### 2.5.6.6 SearchResultsPage

Es ist eine Suchfunktion in der Applikation eingebaut. Die Suchfunktion wird nach dem Standard des neuen Betriebssystems angezeigt indem die sogenannten ‚Charms‘ aufruft. Mit einer Wischbewegung von der rechten Seite des Bildschirms werden diese aufgerufen. Nach Klick auf das Such Symbol kann schon der zu suchende Text eingegeben werden. Die Suchfunktion reagiert immer Kontext abhängig. Das heisst, wenn man innerhalb der StartPage nach einem Begriff sucht, werden nur die „Ongoing“-Objekte durchgesucht. Analog dazu werden in der Ansicht von allen Prozessen oder allen Prozessinstanzen nur die jeweiligen Objekte durchsucht.

### 2.5.6.7 SnappedView

Hier ein Auszug zu SnappedView aus der offiziellen Microsoft Dokumentation:

*[..]Eine angedockte App belegt einen kleinen Bereich des Bildschirms, während eine App in der gefüllten Ansicht den Bildschirmbereich ausfüllt, der nicht von der angedockten App belegt ist. [..] [4]*

Und

**Hinweis** *Angedockte Ansichten und gefüllte Ansichten sind nur für Displays mit einer horizontalen Auflösung von mindestens 1366 relativen Pixeln verfügbar. Und zwar, weil die angedockte Ansicht immer 320 Pixel breit ist und auf beiden Seiten des Bildschirms platziert werden kann. Die verbleibenden 1046 Pixel werden dem Teiler (22 Pixel) und der gefüllten Ansicht zugewiesen, die immer eine horizontale Auflösung von mindestens 1024 relativen Pixeln haben muss. [4]*

Für den Zertifizierungs-Prozess von Windows Store Apps wird eine angedockte Ansicht, snappedView, vorausgesetzt. Was aber dort angezeigt werden muss, ist nicht vorgegeben.

Da die Hauptfunktionalität, das Anzeigen von Objekten in der Master-/Detail-Ansicht, in einer snappedView durch die minimale Breite von 320 Pixeln praktisch unmögliche umzusetzen ist, wird zu keiner Ansicht eine snappedView mit Funktionalität angeboten.

Stattdessen wird immer das Logo des Auftraggebers zu sehen sein, indem bei jeder Änderung der Bildgrösse folgenden Methode aufgerufen wird:

```
private void WindowSizeChanged(object sender, WindowSizeChangedEventArgs e)
{
    ApplicationViewState myViewState = ApplicationView.Value;

    if (myViewState == ApplicationViewState.Snapped
        || myViewState == ApplicationViewState.Filled)
    {
        this.InvalidateVisualState();
        this.Frame.Navigate(typeof(SnappedView));
    }
    else if (myViewState == ApplicationViewState.FullScreenLandscape)
    {
        while (this.Frame.CurrentSourcePageType == typeof(SnappedView))
        {
            this.InvalidateVisualState();
            this.Frame.GoBack();
        }
    }
}
```

**Abbildung 38 WindowSizeChanged EventHandler**

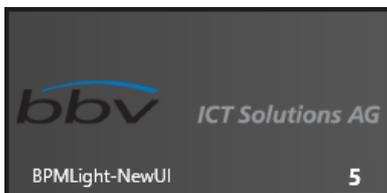
## 2.5.7 Administration

Hier kann der Hintergrund der Applikation festgelegt werden. Aufgrund von technischen Limitationen ist ein Neustart der Applikation nötig um die Änderung zu übernehmen.

## 2.5.8 NotificationExtensions

Dieses Projekt wurde komplett aus einem offiziellen Beispiel-Projekt von Microsoft übernommen. Es hilft dabei, auf der Kachel der Applikation, welche im neuen Start-Bildschirm von Windows 8 angezeigt wird, Informationen anzuzeigen.

In diesem Fall wird aber nur von der Badge-Update Gebrauch gemacht. Dieses Update sendet der Kachel die Anzahl der „Ongoing“-Objekte.



**Abbildung 39 Badge-Update**

Es gäbe noch weitere Möglichkeiten, wie die NotificationExtension benutzt werden können. Zum Beispiel kann anstatt nur einer Zahl auch ein mitgegebener Text dargestellt werden. Zudem kann dieses Projekt dazu verhelfen, ein Symbol der Applikation mit Badge-Update auf dem Lock-Screen (Sperrbildschirm) anzuzeigen. Für diese Möglichkeiten der Benachrichtigungen müsste aber die Applikation stets im Hintergrund ablaufen, weswegen auf einer Implementation während dieser Studienarbeit abgesehen werden musste.

---

## 3. Schlussfolgerung

### 3.1 Zusammenfassung

In dieser Studienarbeit wurden viele Varianten ausgearbeitet, wie eine moderne Tablet-Applikation, eine Windows Store App, auf einem Microsoft Surface RT Tablet aussehen könnte.

Von den anfänglichen Vorstellungen des Studenten, über die Anforderungen des Auftraggebers bis zu den angebotenen Möglichkeiten des Frameworks konnte laufend eine grosse Transition erkannt werden.

Der geeignetste Vorschlag davon wurde so getreu wie möglich oder so viel wie die Zeit erlaubt hat, in einem Prototyp verwirklicht.

Der Prototyp ist im Stande, einen Web Service zu konsumieren und die Daten davon auf ansprechender Weise darzustellen.

Vor allem der Wiederverwendbarkeit von vielen Elementen der Applikation ist grosse Aufmerksamkeit zu schenken.

### 3.2 Beurteilung der Resultate

Da aufgrund des begrenzten Arbeitsaufwandes nur die verlangte Funktionalität des Auftraggebers, das Konsumieren von Daten, verwirklicht werden konnte, bietet diese Studienarbeit eine sehr gute Möglichkeit, diese Applikation später mit mehr Funktionalität auszustatten.

Genauer gesagt, der Windows Store App könnten Funktionen der Datenmanipulation und Datenerstellung hinzugefügt werden, die dann vom Web Server gespeichert werden.

Zudem wäre es möglich, die in den Vorschlägen vorhandenen aufwendigen Applikations-Kacheln zu implementieren. Diese konnten in der Studienarbeit nicht verwirklicht werden, da sie aufgrund der kleinen Auflösung des Ziel-Tablets nicht zur Geltung gekommen wären und zudem die bestehende Infrastruktur dafür nicht vorgesehen war.

Es wäre weiter möglich, die Windows Store App auf normale Desktop-PCs zu installieren. Dafür müssen aber wohl Änderungen im Aufbau vorgenommen werden um die User-Experience zu steigern.

Inwiefern dieser Prototyp auf Smartphones mit Windows Phone 8 als Betriebssystem eingesetzt werden kann und was für Änderungen dafür nötig wären, kann ich keine Auskunft geben.

Wenn man die Verwendung über mehrere Geräte-Typen in Betracht zieht, sollte man auch die Speicherung von Applikations-Daten in die Microsoft-Cloud näher betrachten.

## 4. Persönlicher Bericht

Ich bin an diese Studienarbeit mit sehr viel Eifer und Motivation herangegangen. Die Gründe dafür sind, dass ich mit einer Programmiersprache arbeiten konnte, die mir zusagt und andererseits für mich sehr verlockend war, ein Programm für eine Plattform bzw. für eine Technologie zu entwickeln, die sehr neu ist. Die Plattform für die die Applikation zu entwickeln war, Windows 8 und die neue Kachel-Oberfläche, wurden erst nach der Hälfte der Studienarbeit für die breite Masse zugänglich.

Das heisst für mich als Entwickler, dass man den besten Kompromiss für die Applikation finden musste. Einerseits haben Benutzer keine grobe Vorstellung wie eine Applikation für diese Plattform aussehen sollte und andererseits können sie die Applikation nur verwenden und erleben, wenn sie die Transition zu eben dieser Plattform vollzogen haben. Dann muss man aber bedenken, dass die entwickelte Windows Store App aus dieser Studienarbeit sich nicht zu sehr von den beigelegten Applikationen der Plattform unterscheiden darf. Erschwerend kommt nach dazu, dass die Philosophie der neuen Benutzeroberfläche es von einem verlangt, dass man alles bisherige erlernte vernachlässigt oder ganz vergisst. Es muss schlicht und einfach anders aussehen und bedient werden.

Um diesen Kompromiss zu erreichen, war es nötig, dass der Auftraggeber stets im Entwicklungsprozess der Benutzeroberfläche miteinbezogen werden musste.

Diese Zusammenarbeit hat zu mehr Zeitaufwand geführt aber auch zu einem Dokument, was für die Weiterentwicklung der Applikation sehr hilfreich sein wird.

Um auf die Plattform wieder zurück zu kommen: Ich muss hier andeuten, dass ich mich von der Plattform oft im Stich gelassen gefühlt habe. Je weiter man sich in die Materie eingearbeitet und Erkenntnisse gewonnen hat, desto mehr hat man von den Einschränkungen erfahren.

Diese Einschränkungen haben den Eindruck hinterlassen, dass diese nicht technologisch bedingt sind. Es betraf Fähigkeiten und Möglichkeiten, die einfach ausgelassen oder sogar entfernt worden sind. Ein Anzeichen dafür, dass diese Plattform nicht über alle Zweifel erhaben ist und nicht fertig gestellt ist.

Aber auch die benötigten Entwicklungs-Tools sind oft abgestürzt und haben Probleme verursacht.

Diese Probleme haben aber meinem Eifer und Motivation nicht viel abgewonnen da der Prototyp dennoch verwirklicht werden konnte, auch wenn dafür ein erheblicher Mehraufwand, welcher sich im Projektplan.xlsx widerspiegelt, erwiesen hat.

Ich konnte mich auch an die Freiheiten, die eine Einzelarbeit zulässt, erfreuen. Auch die effizienten Meetings mit dem Betreuer erlaubten eine vorwärts gerichtete Arbeit.

Mit dem Resultat an sich bin ich insofern zufrieden, da ich die Möglichkeit hatte, ein Projekt selber durchführen und mich selbst für die Arbeit motivieren und Anreize finden konnte.

## 5. Literaturverzeichnis

- [1] Microsoft, «.NET for Windows Store apps - supported APIs,» [Online]. Available: <http://msdn.microsoft.com/en-us/library/windows/apps/br230232%28v=VS.85%29.aspx>. [Zugriff am 20 12 2012].
- [2] D. M. Kean, «Dev Center - Windows Store apps,» [Online]. Available: <http://social.msdn.microsoft.com/Forums/en-US/winappswithcsharp/thread/6d7ab585-1872-45fc-a186-d92319ef0cbc/>. [Zugriff am 20 12 2012].
- [3] SmartBear, [Online]. Available: <http://www.soapui.org/>. [Zugriff am 20 12 2012].
- [4] Microsoft, «Richtlinien für angedockte und gefüllte Ansichten (Windows Store-Apps) (Windows),» [Online]. Available: <http://msdn.microsoft.com/de-de/library/windows/apps/hh465371.aspx>. [Zugriff am 11 12 2012].

## 6. Glossar und Abkürzungsverzeichnis

**Tabelle 4 Glossar und Abkürzungsverzeichnis**

<b>Begriff</b>	<b>Beschreibung</b>
<b>.Net</b>	Framework von Microsoft für verschiedene Programmiersprachen
<b>C#</b>	Programmiersprache die auf dem .Net Framework aufbaut
<b>DTO</b>	Data Transfer Object
<b>ICT</b>	Information and communications technology
<b>Mockups</b>	Eine Attrappe um z.B.: Ein UI-Mockup ist ein Entwurf der Benutzeroberfläche
<b>px</b>	Abkürzung für englisches Wort Pixel, deutsch Bildpunkt
<b>RT</b>	Abkürzung für Runtime, bezeichnet eine spezielle Version des Windows Betriebssystems das auf Geräten mit ARM-Prozessoren(Architekturtyp) verwendet wird
<b>SOAP</b>	Simple Object Acces Protocol
<b>UI Prototype</b>	Ein erster Entwurf der Benutzeroberfläche der Applikation.
<b>User Interface GUI, UI</b>	Graphical User Interface - Die Graphische Oberfläche auf der der Benutzer Aktionen ausführen kann.
<b>Webservice</b>	Software-Anwendung die durch das Internet angesprochen werden kann.
<b>WPF</b>	Windows Presentation Foundation
<b>WSDL</b>	Web Services Description Language
<b>XAML</b>	Extensible Application Markup Language, deklarative Programmiersprache

## 7. Abbildungsverzeichnis

Abbildung 1 Architekturübersicht .....	2
Abbildung 2 Startseite SharePoint .....	3
Abbildung 3 Liste von Prozessen .....	3
Abbildung 4 Detail-Ansicht einer Prozessinstanz .....	4
Abbildung 5 Windows Store .....	6
Abbildung 6 Windows Musik.....	7
Abbildung 7 Mockup Startseite .....	8
Abbildung 8 Mockup Listenansicht .....	9
Abbildung 9 Mockup Detail-Ansicht.....	9
Abbildung 10 Startseite .....	10
Abbildung 11 normale Aktuell-Kachel.....	11
Abbildung 12 Timer .....	11
Abbildung 13 Vier-Augen-Prinzip .....	11
Abbildung 14 Grosse und kleine Favoriten-Kachel .....	12
Abbildung 15 Navigations-Kacheln.....	12
Abbildung 16 Alle Prozesse .....	13
Abbildung 17 Grossansicht Prozess-Kachel.....	13
Abbildung 18 Alle Prozessinstanzen.....	14
Abbildung 19 Grossansicht Prozessinstanz-Kachel .....	14
Abbildung 20 Master/Detail-Ansicht.....	15
Abbildung 21 Favoriten Darstellung.....	15
Abbildung 22 Checkliste quittieren .....	16
Abbildung 23 Quittierungs-Bestätigung.....	16
Abbildung 24 Checkliste quittiert.....	17
Abbildung 25 Wiederverwendeter Code.....	18
Abbildung 26 Übernommene Methoden .....	20
Abbildung 27 Hinzugefügte Methoden .....	20
Abbildung 28 Struktur Windows Store App .....	21
Abbildung 29 Common Ordner .....	21
Abbildung 30 Definition ItemTemplate.....	22
Abbildung 31 Favoriten Kachel.....	22
Abbildung 32 Farbdefinition Kachel .....	22
Abbildung 33 Asynchrone Methode.....	23
Abbildung 34 Startseite mit Raster .....	24
Abbildung 35 Detail-Ansicht.....	24
Abbildung 36 Master-/Detail-Ansicht .....	26
Abbildung 37 Quittierungs-Dialog.....	26
Abbildung 38 WindowsSizeChanged EventHandler .....	28
Abbildung 39 Badge-Update .....	28
Abbildung 40 Web-Interface von BPM Light.....	52
Abbildung 41 Windows Store .....	54
Abbildung 42 Listendarstellung aus ‚Reisen‘-Applikation von Windows 8 .....	54
Abbildung 43 Windows Musik.....	55
Abbildung 44 Dashboard aus dem Internet .....	63
Abbildung 45 normale Aktuell-Kachel.....	70
Abbildung 46 Timer .....	71
Abbildung 47 Vier-Augen-Prinzip .....	71
Abbildung 48 Grosse und kleine Favoriten-Kachel .....	71
Abbildung 49 Navigations-Kacheln.....	72
Abbildung 50 Grossansicht Prozess-Kachel.....	73

Abbildung 51 Grossansicht Prozessinstanz-Kachel .....	74
Abbildung 52 Quittierung .....	77

## 8. Skizzenverzeichnis

Skizze 1 Startseite .....	4
Skizze 2 Prozessliste .....	5
Skizze 3 Detail-Ansicht .....	5
Skizze 4 Papier Prototyp .....	53
Skizze 5 Papier-Prototyp Auftraggeber .....	64

## 9. Tabellenverzeichnis

Tabelle 1 Web Service Methoden .....	2
Tabelle 2 ChecklistEntry .....	18
Tabelle 3 ChecklistState .....	18
Tabelle 4 Glossar und Abkürzungsverzeichnis .....	32

## 10. Mockupsverzeichnis

Mockup 1 Variante 1 .....	56
Mockup 2 Variante 1.5 .....	57
Mockup 3 Variante 1.75 .....	57
Mockup 4 Prozessliste .....	58
Mockup 5 Prozessinstanzliste .....	59
Mockup 6 Administration .....	60
Mockup 7 Quittierung in Ampel-Darstellung .....	60
Mockup 8 Variante 2 .....	61
Mockup 9 Quittierung in Ampel-Darstellung .....	61
Mockup 10 Dritter Entwurf .....	65
Mockup 11 Aktuell Kachel haben einen weissen Balken mit Namen der Prozessinstanz .....	67
Mockup 12 Die Farbe der Kachel füllt diese ganz aus. Kein weisser Balken .....	67
Mockup 13 Startseite .....	70
Mockup 14 Alle Prozesse .....	73
Mockup 15 Alle Prozessinstanzen .....	74
Mockup 16 Master/Detail-Ansicht .....	75
Mockup 17 Favoriten Darstellung .....	76
Mockup 18 Checkliste quittieren .....	76
Mockup 19 Quittierung durchgeführt .....	77

# *Projektdokumentation*

## 11. Projektplan.docx

# Projekt: BPMLight

## Projektplan

[Dokumentstruktur basiert auf RUP „Software Development Plan“]

Bejtulla Akiku

## 11.1 Einführung

### 11.1.1 Zweck

Dieses Dokument soll einen Überblick über die Planung und die Durchführung des Projekts BPMLight verschaffen.

### 11.1.2 Gültigkeitsbereich

Die Gültigkeit dieses Projektplans erstreckt sich über das Herbstsemester 2012 im Rahmen einer Studien- bzw. Semesterarbeit.

### 11.1.3 Referenzen

Nachfolgend werden die Dokumente aufgeführt, welche von diesem Dokument referenziert werden:

- *Projektplan.xlsx*

### 11.1.4 Übersicht

Im folgenden Kapitel wird eine Übersicht über das Projekt BPMLight gewährt. Es soll ersichtlich werden was der Zweck und die Ziele des Projekts sind.

Wichtige Teile dieses Dokuments sind die Abschnitte Managementabläufe und Arbeitspakete. Einerseits werden dort der Projektplan, der Zeitplan, die Iterationsplanung und die Meilensteine festgelegt und andererseits werden detaillierte Angaben zu den einzelnen Arbeitspaketen geschildert.

## 11.2 Projekt Übersicht

Die MetroUI-Applikation BPMLight ermöglicht es einem Benutzer den Verlauf und aktuellen Zustand eines Workflows einzusehen. Die Workflows bestehen zum Teil aus Einzelschritten welche durch den Benutzer quittiert werden müssen.

### 11.2.1 Zweck und Ziel

Die Studienarbeit gibt dem Studenten die Möglichkeit, die erworbenen Fähigkeiten und Kenntnisse aus den Vorlesungen und Übungen anzuwenden. Es soll zudem aufzeigen, dass der Student in der Lage ist, ein Problem selbstständig und unter der Anwendung von geeigneten Methoden zu bewältigen.

Der Auftraggeber der Arbeit BPMLight ist die in Zug ansässige Firma bbv ICT Solutions AG unter der Leitung von Herr Adrian Bachofen.

Die zu entwickelnde Applikation BPMLight soll schnell und einfach zu bedienen sein.

### 11.2.2 Annahmen und Einschränkungen

Die Applikation wird hauptsächlich von nur einem Typ Benutzer verwendet. Der sogenannte Controller überwacht und hat den Überblick über die Workflows. Die Software wird deswegen auf ihn abgestimmt.

Die Entwicklung wird zudem nur auf einem Touchscreen fähigen Tablet mit Windows 8 als Betriebssystem entwickelt. Alle anderen Möglichkeiten wie Desktop-Benutzung oder Benutzung auf Handy mit Windows Phone 8 werden nicht berücksichtigt.

Aus Technologischer Sicht werden für die Entwicklung die Programmiersprachen C# und XAML verwendet. Dies schränkt vielleicht die optischen Möglichkeiten (Animationen und GUI-Elemente) ein, ist aber leichter zu implementieren und zu verwalten.

### 11.2.3 Kein System.Data Namespace – Keine lokale Datenbank

Für Windows Store Apps steht der Namespace System.Data nicht zu Verfügung. Diese führt dazu, dass keine lokalen Datenbanken verwendet werden können.

### 11.2.4 Kein WSHttpBinding – Kein SOAP 1.2

Windows Store Apps unterstützen bei Beginn dieser Studienarbeit kein Web Service vom Typ SAOP 1.2. Es soll auf den älteren Standard, SAOP 1.1, zurückgegriffen werden.

## 11.4 Projektorganisation

### 11.4.1 Organisationsstruktur

Die komplette Arbeit wird durch Bejtulla Akiku durchgeführt.

### 11.4.2 Externe Schnittstellen

<b>Funktion</b>	<b>Name</b>
Betreuer	Prof. Hansjörg Huser
Auftraggeber	Adrian Bachofen

## 11.5 Management Abläufe

### 11.5.1 Projekt Kostenvoranschlag

Bis zu 14 Wochen sind geplant für das Projekt. Nach 13 Wochen ist das Projekt jedoch beendet wobei die letzte Woche als Reserve freigehalten wird

### 11.5.2 Projektplan / Zeitplan

Der detaillierte Projektplan (inklusive Zeitplanung) ist unter *Projektplan.xlsx* ersichtlich.

### 11.5.3 Iterationsplanung

Die Inception Phase umfasst zwei Wochen.

Im Anschluss an die Inception Phase folgt eine Iteration der Elaboration Phase von zwei Wochen.

Anschließend folgen zwei Iterationen der Construction Phase:

- Construction (1. Iteration): Zwei Wochen.
  - Prototyp erstellt
- Construction (2. Iteration): Zwei Wochen.
  - Feedback von Auftraggeber umsetzen

Darauf folgt die Transition Phase, welche eine Wochen umfasst und das Projekt abschliesst.

Vorsichtshalber wird nach der Transition Phase noch eine Woche Reserve eingeplant, wo noch allfällige Probleme gelöst bzw. ungenügende Teile verbessert werden können.

### 11.5.4 Meilensteine

Nachfolgend werden die geplanten Meilensteine aufgeführt

Datum	Meilenstein	Phase	Ziele / Resultate
17.09.2012	Start	Inception (W1)	Projektstart
24.09.2012	Aufgabenstellung, Projektplan.xlsx	Inception (W2)	Beim Meeting mit Auftraggeber wurde die Aufgabenstellung vermittelt. Der Projektplan.xlsx wird mit diesen Informationen erstellt
01.10.2012	Mockups Projektplan.docx	Elaboration (W3)	Drei verschiedene Mockups für Meeting mit Auftragsgeber erstellt. Projektplan.docx erstellt
8.10.2012	Meeting mit Auftragsgeber	Elaboration (W4)	Die Mockups wurden vom Auftraggeber bewertet. Das Design sollte zu 90% bestimmt sein.
15.10.2012	Beginn Construction 1	Construction 1 (W5)	
4.11.2012	Ende Construction 1	Construction 1 (W7)	Prototyp von BPMLight kann dem Auftraggeber gezeigt werden
5.11.2012	Demo Prototyp	Construction 2 (W8)	
5.11.2012	Beginn Construction 2. Iteration	Construction 2 (W8)	
9.12.2012	Ende Construction 2.	Construction 2	Endprodukt fertig gestellt

---

Datum	Meilenstein	Phase	Ziele / Resultate
	Iteration	(W12)	
10.12.2012	Transition beginnt	Transition (W13)	

---

### 11.5.5 Besprechungen

Der Student trifft sich wöchentlich mit dem Betreuer Prof. H. Huser jeweils am Donnerstag um 11h an der HSR.

Zusätzlich werden bei Bedarf Meetings mit dem Auftraggeber abgehalten.

### 11.5.6 Abgaben

Nachfolgend werden die geplanten Releases der Software aufgeführt:

- **Architektur-Prototyp** – Proof of Concept der Metro-Applikation
- **GUI Prototyp** – Mindestens drei verschieden UI-Mockups in Metro-Style.
- **Beta Release** – Beta-Version des Endproduktes mit den primären Funktionalitäten.

## 11.6 Arbeitspakete

Nachfolgend findet sich eine Auflistung sämtlicher Arbeitspakete. Dabei werden folgende Fragen geklärt:

- Wie heisst das Arbeitspaket?
- Was beinhaltet es?
- Wie viel Zeit wird dafür benötigt?
- Gibt es Abhängigkeiten zu anderen Arbeitspaketen?
- Gibt es spezielle Risiken oder Probleme, die beachtet werden müssen?

Detaillierte Aufwandschätzungen sind im *Projektplan.xlsx* ersichtlich, welcher auch die Zeitplanung für die einzelnen Komponenten darlegt.

### 11.6.1 User Interface

#### 11.6.1.1 Beschreibung

Das User Interface soll sich an die Guidelines und die Philosophie der Metro-UI von Microsoft halten. Es wird somit mit vielen farbigen Kacheln gearbeitet und soll trotzdem schlicht und leichtgewichtig daherkommen.

Das flüssige Interagieren mit dem UI soll immer gewährleistet sein.

#### 11.6.1.2 Zeitaufwand

Um die verschiedenen Designs zu erstellen welche dem Auftraggeber gefallen, muss viel Zeit aufgewendet werden. Jedoch wird weiterhin viel Zeit für das UI gebraucht da erst bei der Construction-Phase die verfügbaren UI-Element und die Möglichkeiten erkannt werden, welche das Framework zur Verfügung stellt.

Dieses Arbeitspaket beinhaltet auch die Navigation der Applikation.

Die Aufwandschätzungen sind im *Projektplan.xls* aufgeführt.

#### 11.6.1.3 Risiken

Das UI lässt sich in XAML programmieren und stellt somit kein Problem dar. Die Einigung auf ein geeignetes Design mit dem Auftraggeber könnte unter Umständen zu weiteren zeitlichen Aufwendungen führen.

### 11.6.2 Use Case

Die Use-Cases für die Metro-Applikation müssen ausgearbeitet werden.

#### 11.6.2.1 Beschreibung

BPMLight existiert soweit als eine SharePoint-Implementation. Die Use-Cases die dort verwendet werden, werden nicht alle für die Metro-Applikation gebraucht. Da das UI sich von der SharePoint-Implementation grundlegend unterscheidet, müssen die Use-Cases spezifisch auf die Metro-Applikation ausgelegt werden.

#### 11.6.2.2 Zeitaufwand

Die Aufwandschätzungen sind im *Projektplan.xls* aufgeführt.

#### **11.6.2.2.1 Risiken**

Die Anzahl der Use-Cases können evtl. vom Auftraggeber erhöht werden.

### **11.6.3 Prototyp**

#### **11.6.3.1 Beschreibung**

Bis zur neunten Projektwoche soll ein erster lauffähiger Prototyp dem Auftraggeber übergeben werden können.

#### **11.6.3.2 Zeitaufwand**

Die Aufwandschätzungen sind im *Projektplan.xls* aufgeführt.

#### **11.6.3.3 Risiken**

Der Funktionsumfang des Prototypen darf nicht zu gross sein.

## 11.7 Infrastruktur

### Entwicklungs Tools

- Microsoft Visual Studio 2012
- SQL Datenbank Server
- GIT

### Geräte

- Workstations für Entwicklung
- Windows-Tablet mit Windows 8 als Zielplattform

### Dokumentation

- Microsoft Office 2010
- [www.fluidui.com](http://www.fluidui.com) – Online Mockup Tool

## 11.8 Qualitätsmassnahmen

Folgende Tabelle gibt einen Überblick über die einzelnen Qualitätsmassnahmen und deren Ziele. Einzelne Massnahmen sind während dem ganzen Projekt relevant, andere nur in bestimmten Phasen.

<b>Massnahme</b>	<b>Zeitraum</b>	<b>Ziel</b>
Fortlaufende Zeiterfassung	ganzes Projekt	ständiger Ist / Soll Abgleich mit Projektplan
Coding Style Guidelines (Naming, Format)	Elaboration 2, kurz vor Entwicklungsbeginn	Einheitliche Terminologie, hohe Lesbarkeit
User Centered Design	Inception bis Construction	Erkennen von Benutzeranforderungen, verhindern von unnötigem Aufwand
Bewährte PM und SE Techniken anwenden. So zum Beispiel RUP, UML und Patterns	ganzes Projekt	Aus Erfahrungen anderer lernen und damit effizienter arbeiten. Qualitätsniveau erhöhen.

## 12. Projektplan.xlsx

Detailplan

BPMLight

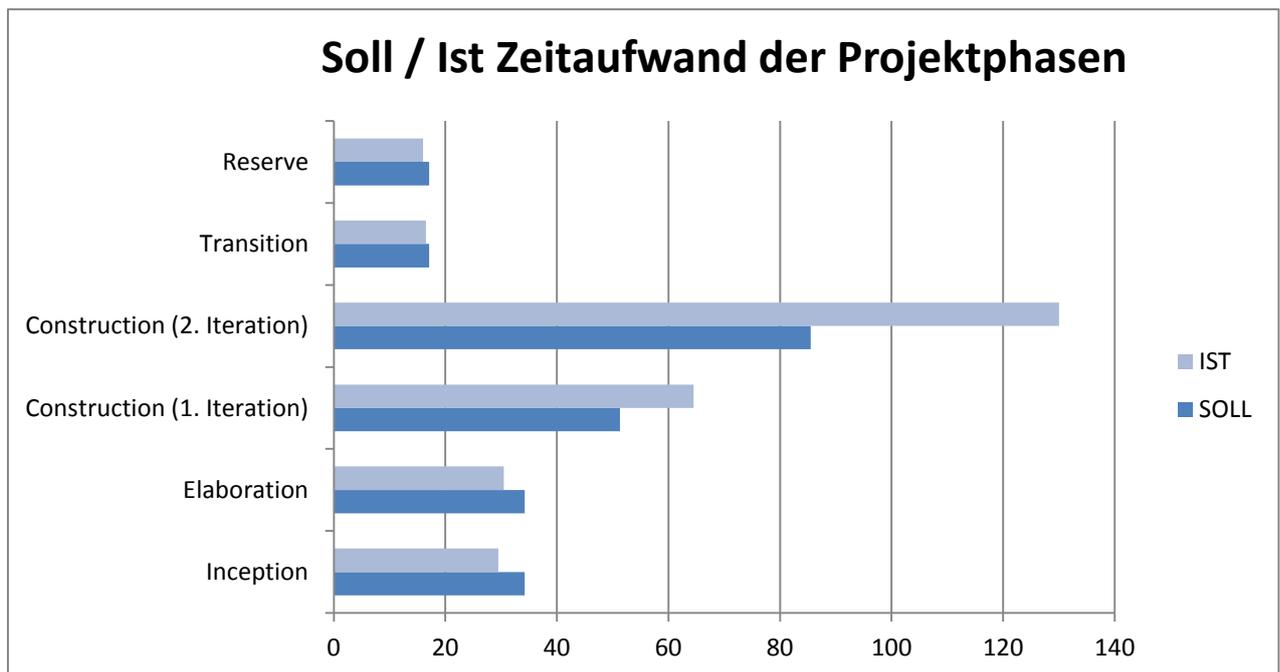
			Inception		Elaboration			Construction (1. Iteration)			Construction (2. Iteration)			Transition	Reserve			
Phasen																		
Meilensteine																		
Meilensteine (weitere)			17.09							05.11							21.12	Abgabe
			Applikation Prototyp															
Woche (von)			W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W12	W14		
			17.09	24.09	01.10	08.10	15.10	22.10	29.10	05.11	12.11	19.11	26.11	03.12	10.12	17.12		
Task	Plan	Effektiv																
	SOLL	IST	I	I	I	I	I	I	I	I	I	I	I	I	I	I		
Projektplan	12.0	10.7	0.0	2.7	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0		
Erstellung	8.0	8.7		2.7	6.0													
Überarbeitung	4.0	2.0														2.0		
Analyse	4.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
Use Cases	4.0	1.0			1.0													
Einarbeitung BPMLight (SharePoint)	12.0	7.5	0.0	0.0	0.0	0.0	6.0	0.5	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0		
Klassenmodell	6.0	3.0					3.0											
Systemarchitektur	2.0	3.0					1.5	0.5			1.0							
Schnittstellendesign	2.0	1.5					1.5											
Ablaufdiagramme (UML)	2.0	0.0																
Implementation	70.0	118.5	0.0	0.0	0.0	0.0	0.0	10.5	20.5	30.0	9.0	18.0	21.0	9.5	0.0	0.0		
Mobile Client	70.0	118.5	0.0	0.0	0.0	0.0	0.0	10.5	20.5	30.0	9.0	18.0	21.0	9.5	0.0	0.0		
User Interface	46.0	54.5							9.0	24.0	1.0	6.0	6.5	8.0				
Business Logic	24.0	42.0								6.0	8.0	12.0	14.5	1.5				
Web Service	18.0	22.0						10.5	11.5									

Studienarbeit: BPMLight - NewUI

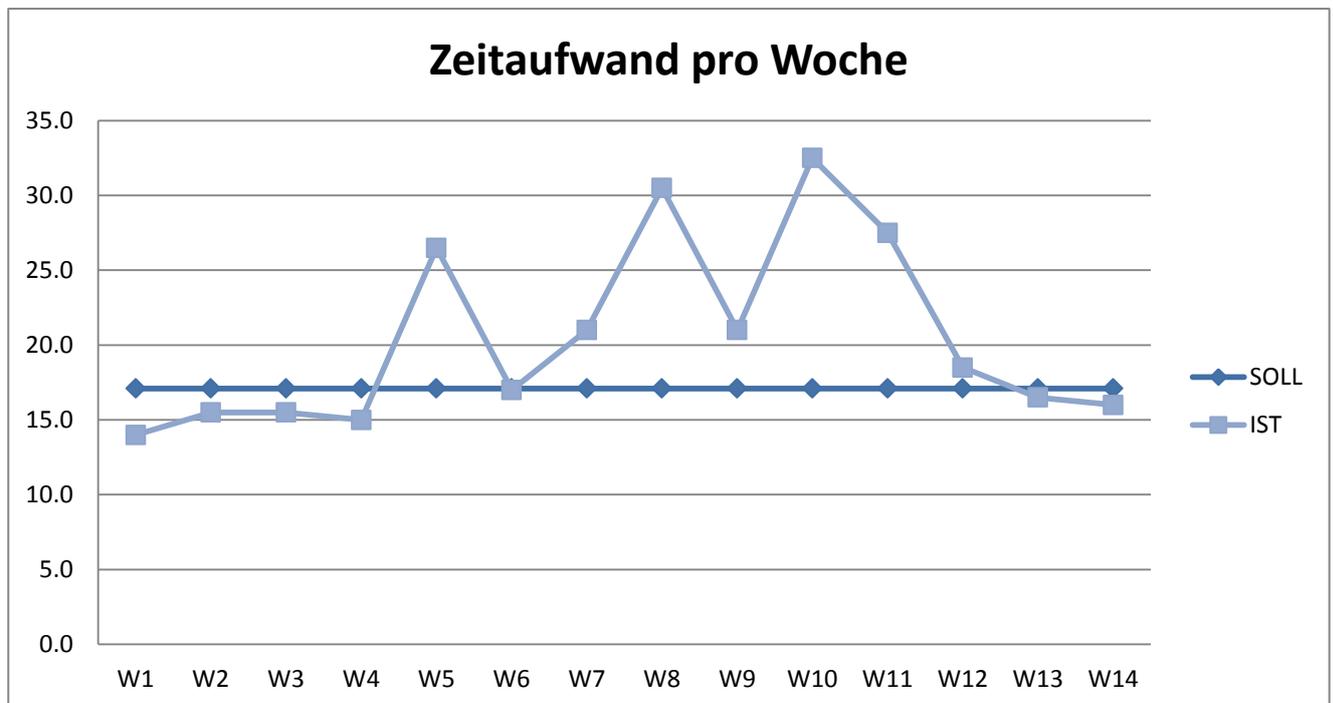
Test	8.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
Integrationstest	8.0	3.0										3.0					
Dokumentation	37.0	57.5	0.0	6.0	7.0	2.0	17.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.0	9.0
Design	16.0	32.5		6.0	7.0	2.0	17.5										
Technischer Bericht und sonstige Abgaben	21.0	25.0														16.0	9.0
<b>Tasks wiederkehrend</b>																	
Studium Technologien	32.0	34.0	6.0	1.5	1.0	11.0	2.0	3.5	0.0	0.0	5.0	0.0	3.0	1.0	0.0	0.0	0.0
Metro	20.0	17.5	6.0	1.5	1.0	9.0											
GIT	6.0	2.0				2.0											
WSDL	6.0	14.5					2.0	3.5			5.0		3.0	1.0			
Sitzungen	14.0	12.0	0.5	4.5	0.5	2.0	0.5	1.0	0.5	0.5	0.0	0.5	0.5	0.5	0.5	0.5	0.0
Wöchentliche Sitzungen	14.0	5.0	0.5	0.5	0.5	0.0	0.5	0.0	0.5	0.5	0.0	0.5	0.5	0.5	0.5	0.5	0.0
Meilensteinsitzung		7.0		4.0		2.0		1.0									
Qualitätssicherung	24.0	31.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	14.0	3.0	7.5	0.0	5.0	
Review Code	10.0	21.0									2.0	14.0	3.0	2.0			
Review Dokumente	14.0	10.5												5.5		5.0	
Sonstiges	27.0	11.3	7.5	0.8	0.0	0.0	0.5	1.5	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Zeiterfassung (im Projektplan)	3.5	0.0															
Protokolle	3.5	2.3	1.5	0.8													
Sonstiges	20.0	9.0	6.0				0.5	1.5			1.0						
<b>Wochentotal</b>				14.0	15.5	15.5	15.0	26.5	17.0	21.0	30.5	21.0	32.5	27.5	18.5	16.5	16.0
<b>Total</b>	<b>240.0</b>	<b>287.0</b>															

Auswertung:

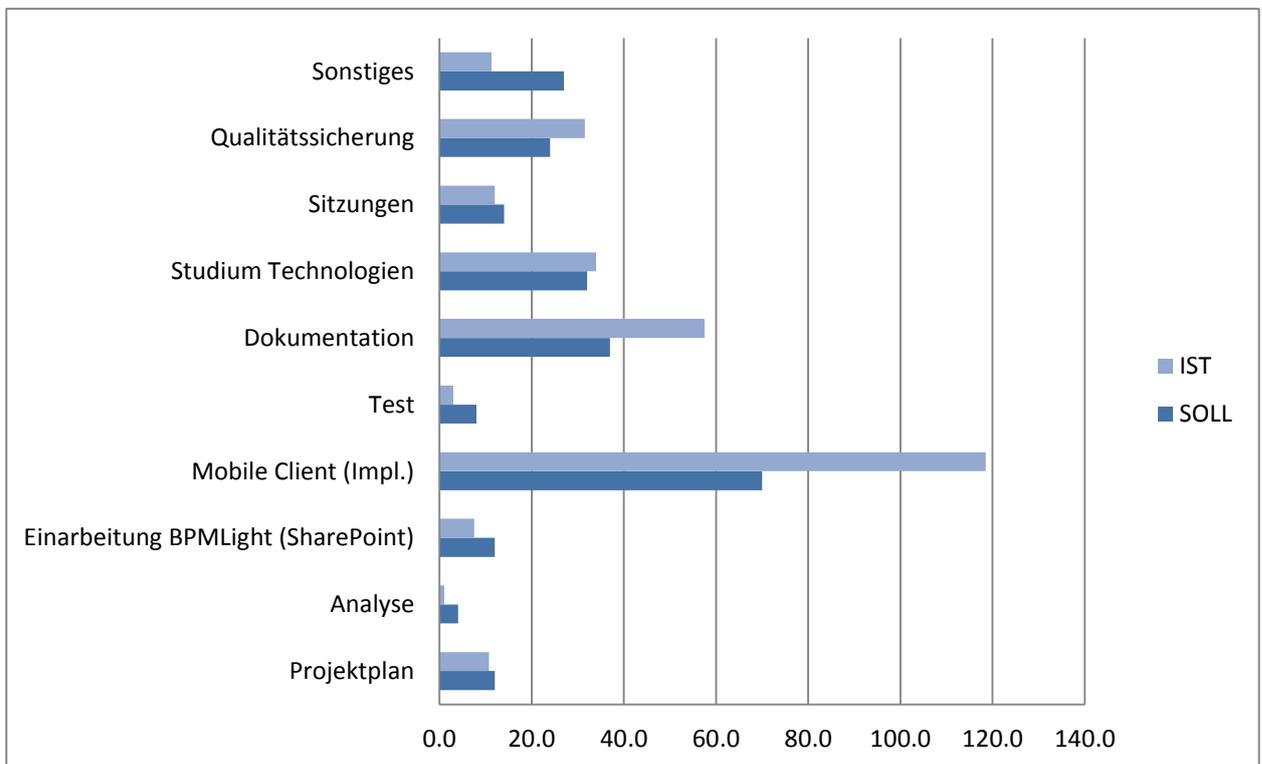
Phasen	Gesamt	
	SOLL	IST
Inception	34.2	29.5
Elaboration	34.2	30.5
Construction (1. Iteration)	51.3	64.5
Construction (2. Iteration)	85.5	130.0
Transition	17.1	16.5
Reserve	17.1	16.0
<b>TOTAL</b>	<b>240.0</b>	<b>287.0</b>



Zeitaufwand pro Woche	SOLL	IST
W1	17.1	14.0
W2	17.1	15.5
W3	17.1	15.5
W4	17.1	15.0
W5	17.1	26.5
W6	17.1	17.0
W7	17.1	21.0
W8	17.1	30.5
W9	17.1	21.0
W10	17.1	32.5
W11	17.1	27.5
W12	17.1	18.5
W13	17.1	16.5
W14	17.1	16.0



Zeitaufwand pro Task	SOLL	IST
Projektplan	12.0	10.7
Analyse	4.0	1.0
Einarbeitung BPMLight (SharePoint)	12.0	7.5
Mobile Client (Impl.)	70.0	118.5
Test	8.0	3.0
Dokumentation	37.0	57.5
Studium Technologien	32.0	34.0
Sitzungen	14.0	12.0
Qualitätssicherung	24.0	31.5
Sonstiges	27.0	11.3



## 13. Mockup.docx

# Projekt: BPMLight

## Mockups

Bejtulla Akiku

## 13.1 Einführung

Dieses Dokument enthält die Entstehungsgeschichte der Mockups für das Projekt BPM Light

## 13.2 Erster Entwurf

Nach Betrachtung der SharePoint-Implementation von BPM Light und ohne weitere Absprache mit dem Auftraggeber wurde ein erstes Mockup erstellt.

Dabei habe ich mir stets die Frage gestellt, „wie kann man die Funktionen von der bestehenden Implementation in einem Metro-orientiertem Design umsetzen?“

### BPM Light - SharePoint

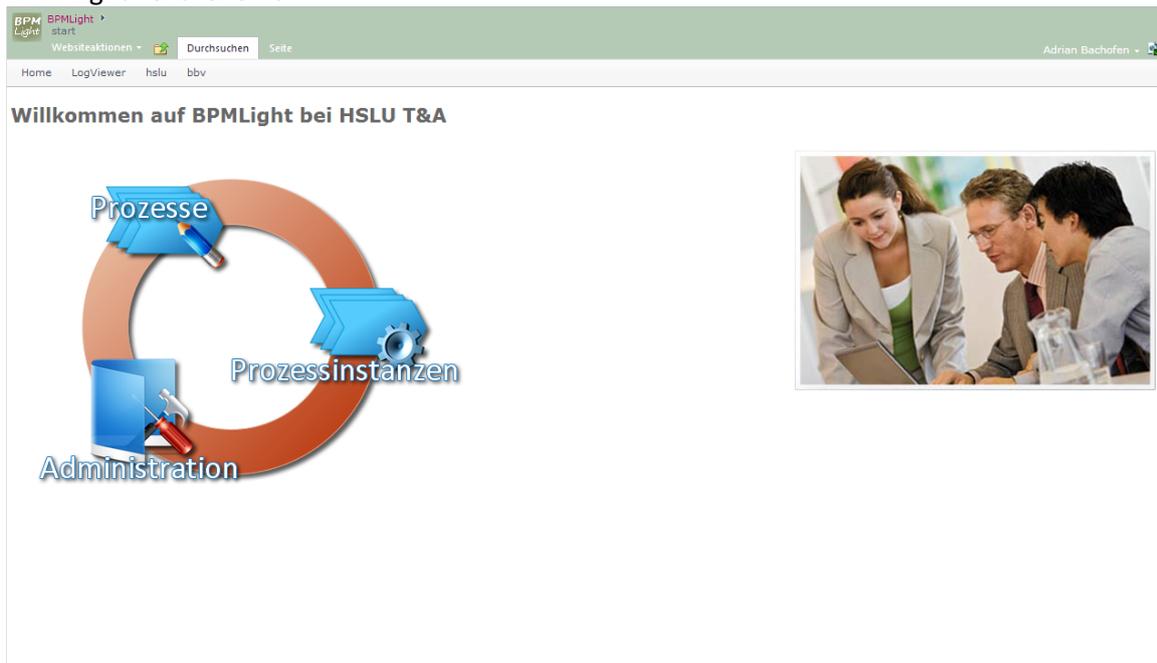
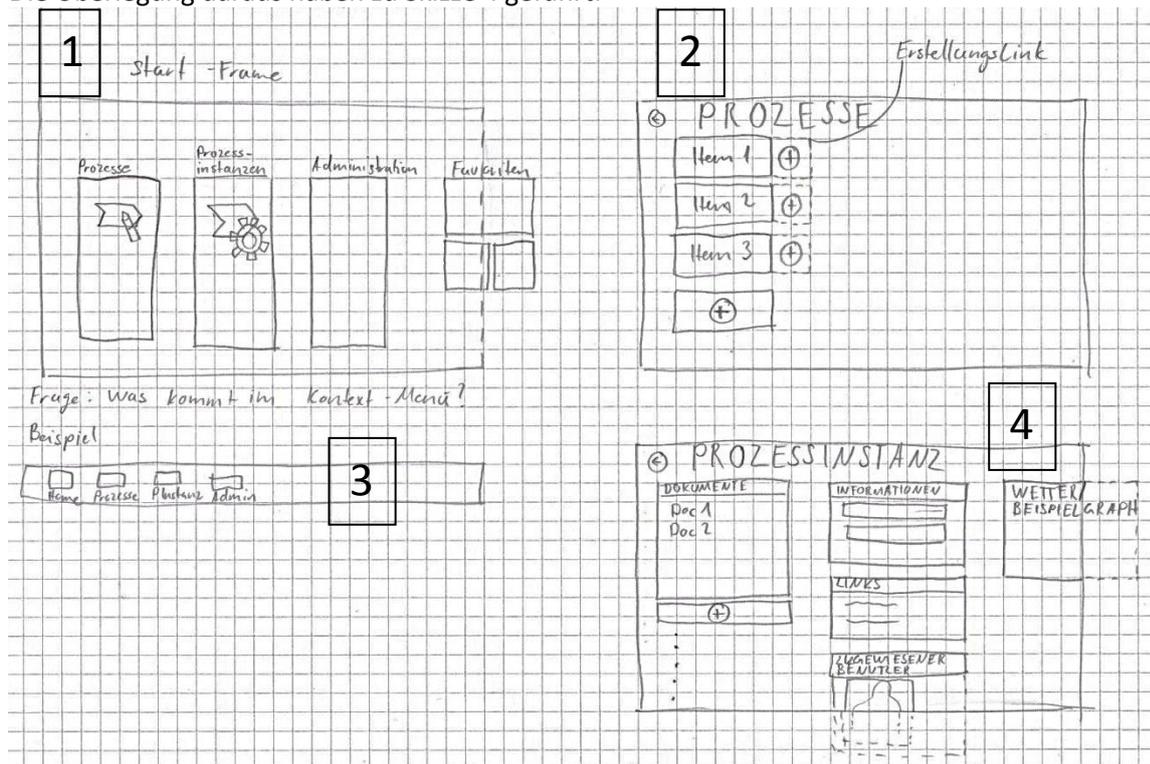


Abbildung 40 Web-Interface von BPM Light

Die Überlegung daraus haben zu Skizze 4 geführt:



#### Skizze 4 Papier Prototyp

1. Der Start-Frame ist enthält grundsätzlich die gleiche Funktionalität wie die SharePoint-Implementation. Man kann von hier aus nach ‚Prozesse‘, ‚Prozessinstanzen‘ und ‚Administration‘ navigieren.  
Zusätzlich werden noch Favoriten vom Benutzer angezeigt. Diese könnte entweder nach Häufigkeit des Aufrufes anordnen oder der Benutzer erstellt diese selber.
2. Alle Prozesse werden als Metro-Kachel dargestellt. Von hier aus kann sofort auf den Erstellungslink eines Prozess geklickt werden um eine neue Prozessinstanz zu erstellen. Um einen neue Prozess zu erstellen, kann auf die unterste Kachel mit dem Plus geklickt werden.
3. Der übliche Rechtsklick einer Maus bringt bei Metro-Applikationen einen Balken mit verschiedenen Links hervor. Hier könnte man die folgenden Links einfügen:
  - a. Home
  - b. Prozesse
  - c. Prozessinstanzen
  - d. Administration

Diese Idee muss aber verworfen werden da man zu wenig Berücksichtigt hat, dass die Applikation mit den Fingern bedient werden soll. Als Ersatz könnte hier eine Pinch-To-Zoom Geste verwendet werden.

4. Die in der SharePoint-Implementation beinhaltenden Informationen könnte in Metro in dieser Art abgebildet werden.

Generell kann man sage, dass die Elemente und das Layout von den Metro Applikationen inspiriert sind, welche standardmässig mit Windows 8 ausgeliefert werden.

#### 13.2.1 Metro Beispiele

Als Beispiel für grosse, aussagekräftige Kacheln kann man sich am Windows Store orientieren.

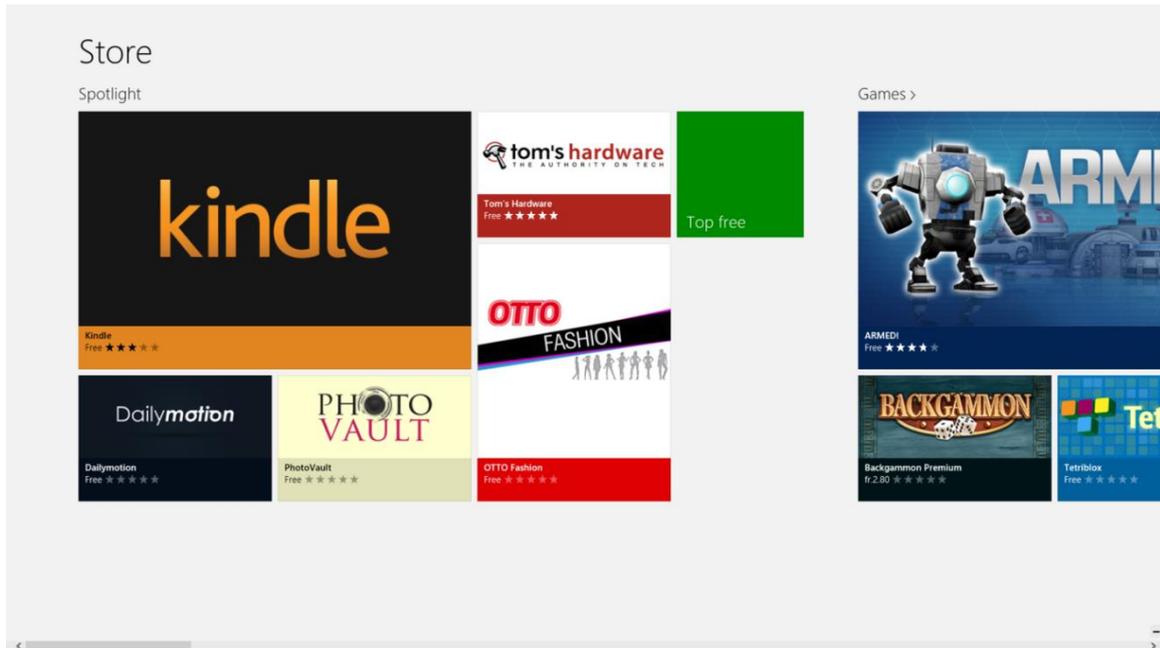


Abbildung 41 Windows Store

Die Vorteile und Nachteile von diesem Layout:

**Vorteile:**

- **Anpassbarkeit:** Die Hauptfunktionalitäten können gross abgebildet und Favoriten können auf den Start-Frame gezogen werden. Erlaubt schnellen Einstieg.
- **Grosse Bilder und Schriften:** man braucht das Surface Tablet nicht näher an sich ziehen um den Inhalt lesen zu können.
- **Fingerfreundlich**

**Nachteile:**

- **Schwierig, aussagekräftige Bilder zu finden**

Da bei BPMLight viele Listen angezeigt werden, könnte man sich an diesen Elementen von Abbildung 42 bedienen:

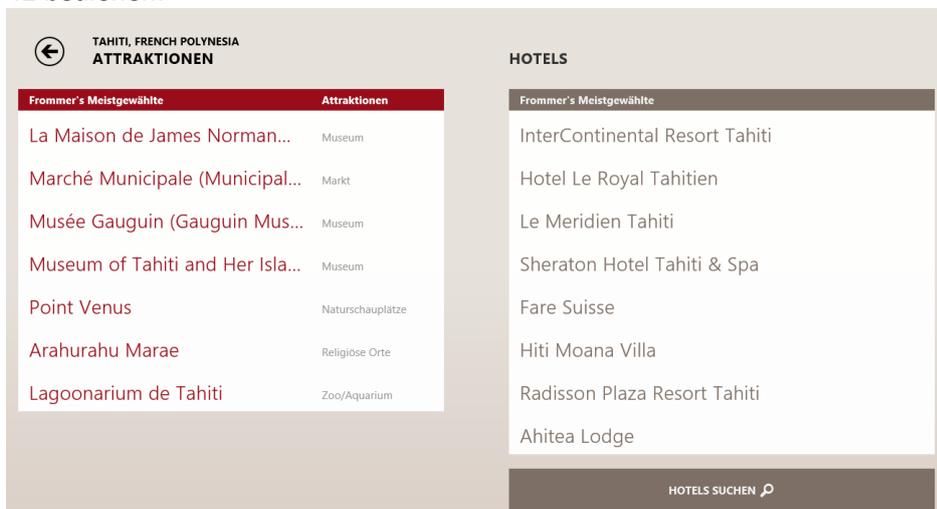


Abbildung 42 Listendarstellung aus ‚Reisen‘-Applikation von Windows 8

Als Gegenbeispiel für ein sachlich orientiertes Layout. Die Musik Applikation von Windows 8:

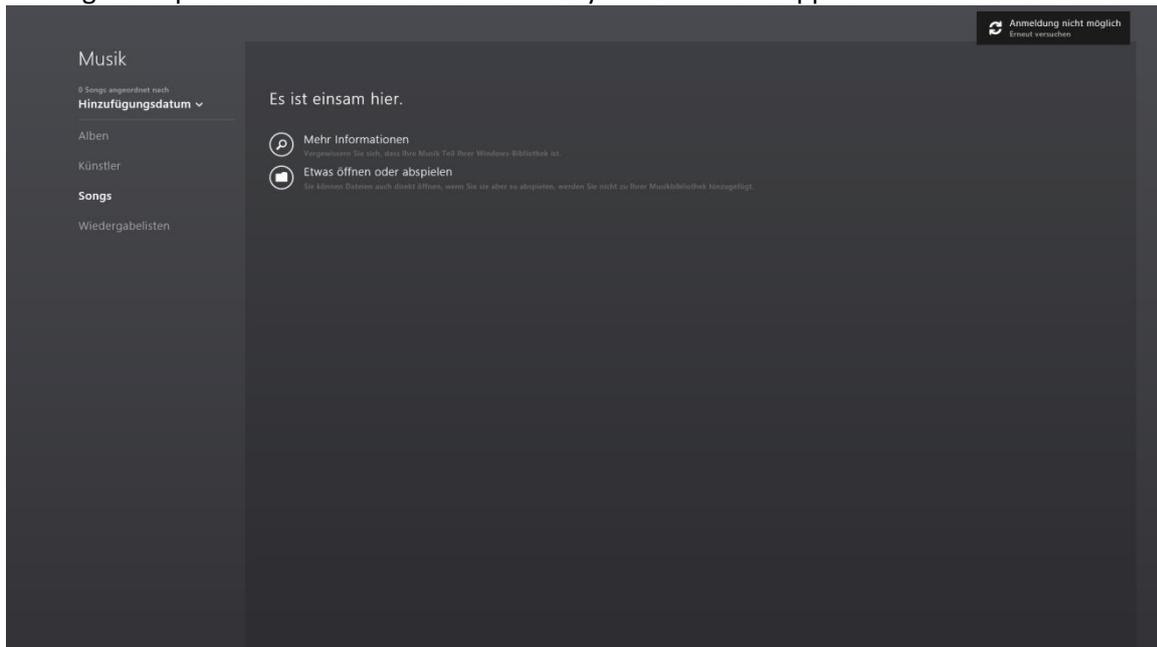


Abbildung 43 Windows Musik

#### Vorteile

- **Navigation links sehr übersichtlich**
- **Layout rechts bietet viel Platz → weniger scrollen**

#### Nachteil

- **Nichts aufregendes**

### 13.2.2 Resultat

Beim Meeting mit dem Auftraggeber vom 8. Oktober 2012 in Zug wurde die oben dargestellte Skizze vom Auftraggeber als unverständlich bezeichnet und die fehlenden Farben wurden bemängelt. Die Applikation müsse auf den Benutzer ‚Controller‘ abgerichtet werden. Ein ‚Controller‘ möchte stets den Zustand eines Workflows verfolgen indem ihm die abgeschlossenen Checklisten eines Workflows zur Begutachtung und Quittierung vorgelegt werden.

Zudem wäre es wichtig, die Berechtigung eines Controllers zu berücksichtigen und ihm nur die Prozesse und Prozessinstanzen angezeigt, die ihn betreffen.

### 13.3 Zweiter Entwurf

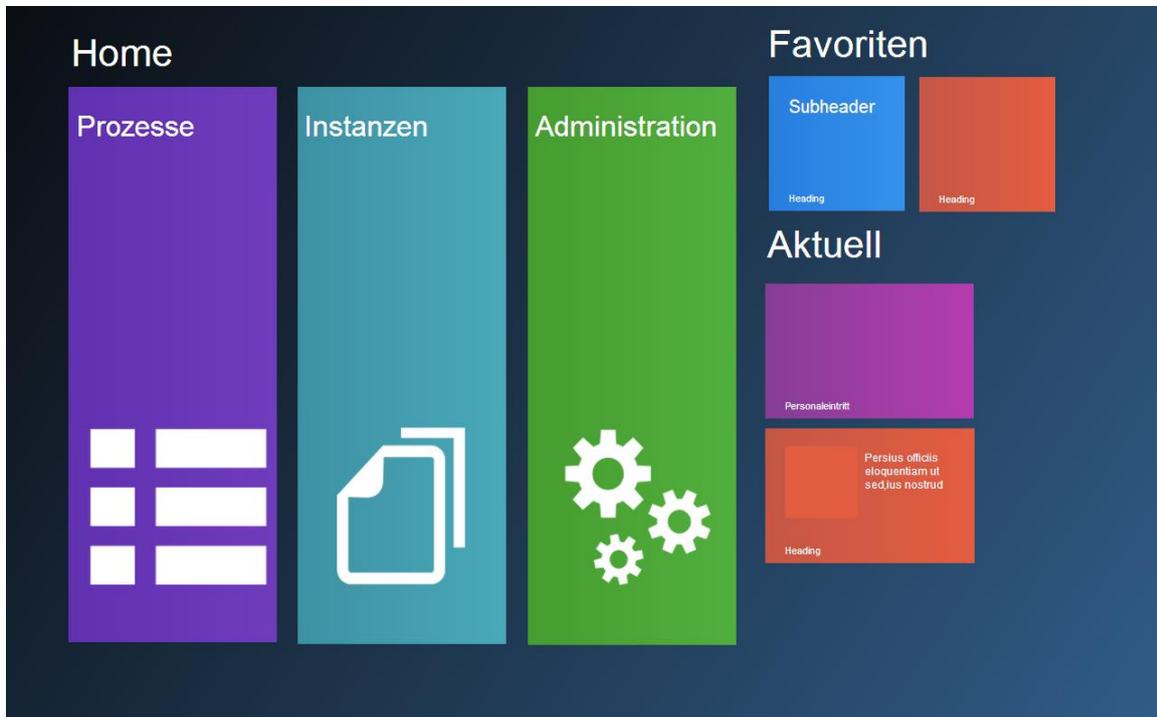
Das Betriebssystem Windows 8 wird erst am 26. Oktober für die Öffentlichkeit verfügbar sein. Aufgrund dieses frühen Stadiums der Verbreitung der Technologie und UI hat es zuweilen keine ausgereiften Mocking-Tools um Mockups zu erstellen.

Nach langer Recherche wurde aber ein Tool<sup>i</sup> gefunden, welches für den Anfang reichen sollte. Zwar hat dieses Tool diverse Einschränkungen und Fehler, aber das Look&Feel von Windows 8 Metro Applikationen kann vermittelt werden.

Bei den folgenden Varianten ging es darum, der farblosen Skizze den Look&Feel von Metro zu verleihen. Es wurde somit bewusst mit vielen Farben gearbeitet.

Generell muss man sagen, dass eine Kachel in Metro einen Link verkörpert. Diese Kacheln können ‚Live Informationen‘ beinhalten welche bei Bedarf im Sinne einer Notifikation angezeigt werden.

### 13.3.1 Variante 1



#### Mockup 1 Variante 1

- Der grösste Platz wird von den Funktionen eingenommen, welche in der SharePoint-Implementation vorhanden sind.
  - ‚Prozesse‘ verlinkt auf eine Liste von allen Prozessen
  - ‚Instanzen‘ verlinkt auf eine Liste von allen Prozessinstanzen
  - ‚Administration‘ verlinkt auf alle Pendenzen
- Grosse Kacheln und Icons
  - Führt zu guter Lesbarkeit was bei einer Tablet Applikation sehr wichtig ist
- Bietet kleines Mass an Individualisierbarkeit indem der Benutzer Prozesse oder Prozessinstanzen welche er verfolgen will als Favoriten auf den Homescreen anheften können.
  - Durch horizontales scrollen können beliebig viele Favoriten angezeigt werden.
- Pendenzen werden unter „Aktuell“ angezeigt. Unter Pendenzen versteht man Checklisten, welche vom Controller quittiert werden müssen
- Icons sind Metro-konform

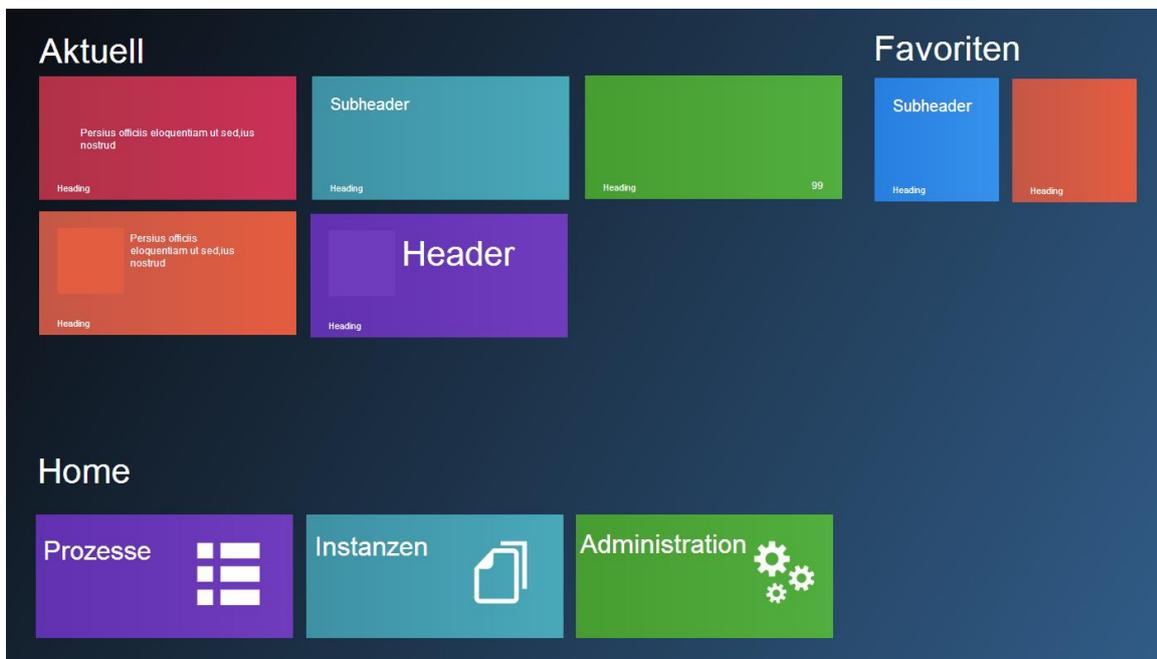
### 13.3.2 Variante 1.5



#### Mockup 2 Variante 1.5

- Grundidee wie oben
- Die Kacheln der SharePoint-Funktionen sind kleiner dargestellt.
- Unter Aktuell nehmen die Pendenzen mehr Platz ein um mehr Aufmerksamkeit zu erregen.

### 13.3.3 Variante 1.75



#### Mockup 3 Variante 1.75

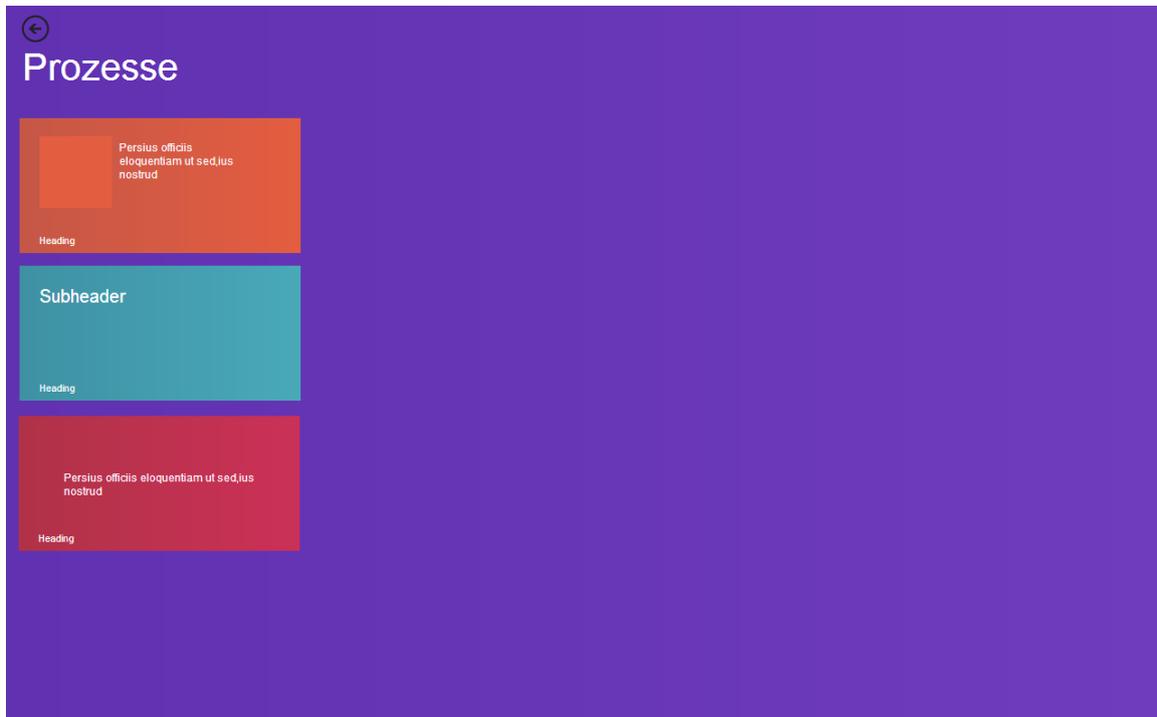
- Die Kacheln von aktuell nehmen am meisten Platz ein und stellen somit auf den ersten Blick am meisten Informationen dar
- Bei den Favoriten hat sich nichts verändert

- Unter Home ist die Navigation zu verstehen.

### 13.3.3.1 Folgescreens vom zweiten Entwurf

Folgend sind die Screens dargestellt, welche bei Klick auf eines der Home-Kacheln erscheinen. Dabei wird die Farbe der Kachel aus dem Homescreen als Hintergrund des neuen Screens gewählt.

### 13.3.3.2 Prozesse

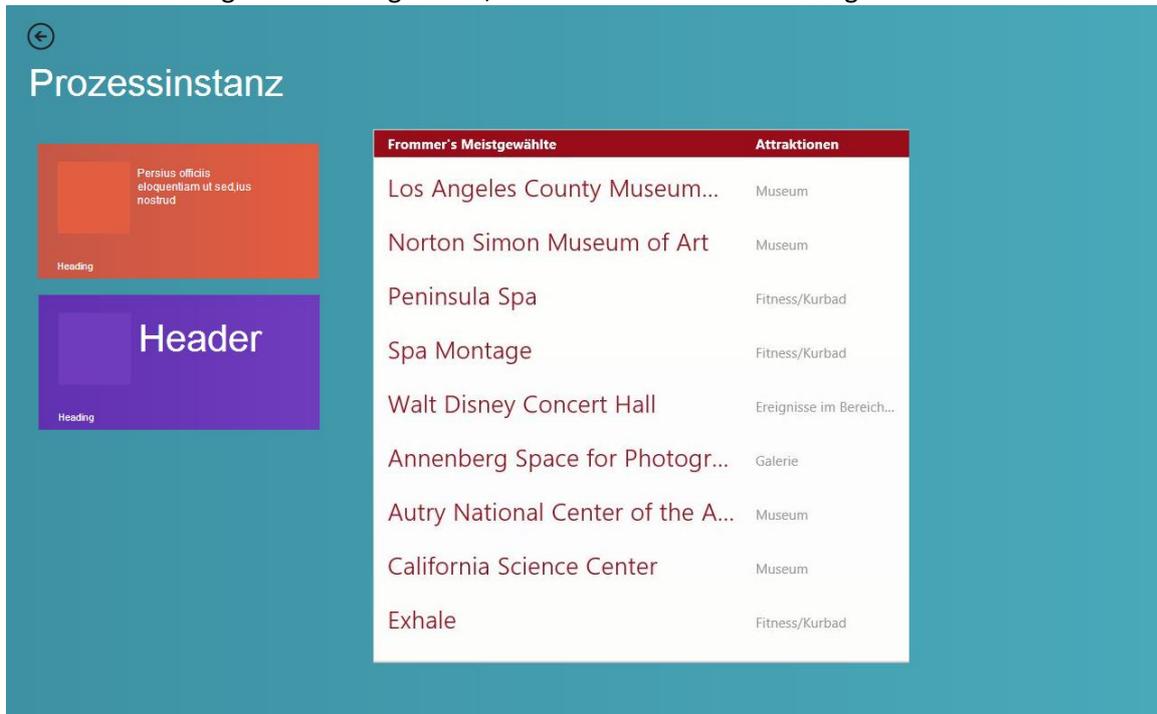


#### Mockup 4 Prozessliste

- Für jeden Prozess eine eigene Kachel
- Die drei verschieden aufgebauten Kacheln können diverse Informationen anzeigen

### 13.3.3.3 Prozessinstanzen

Hier sind zwei Möglichkeiten abgebildet, wie die Prozessinstanzen dargestellt werden können.

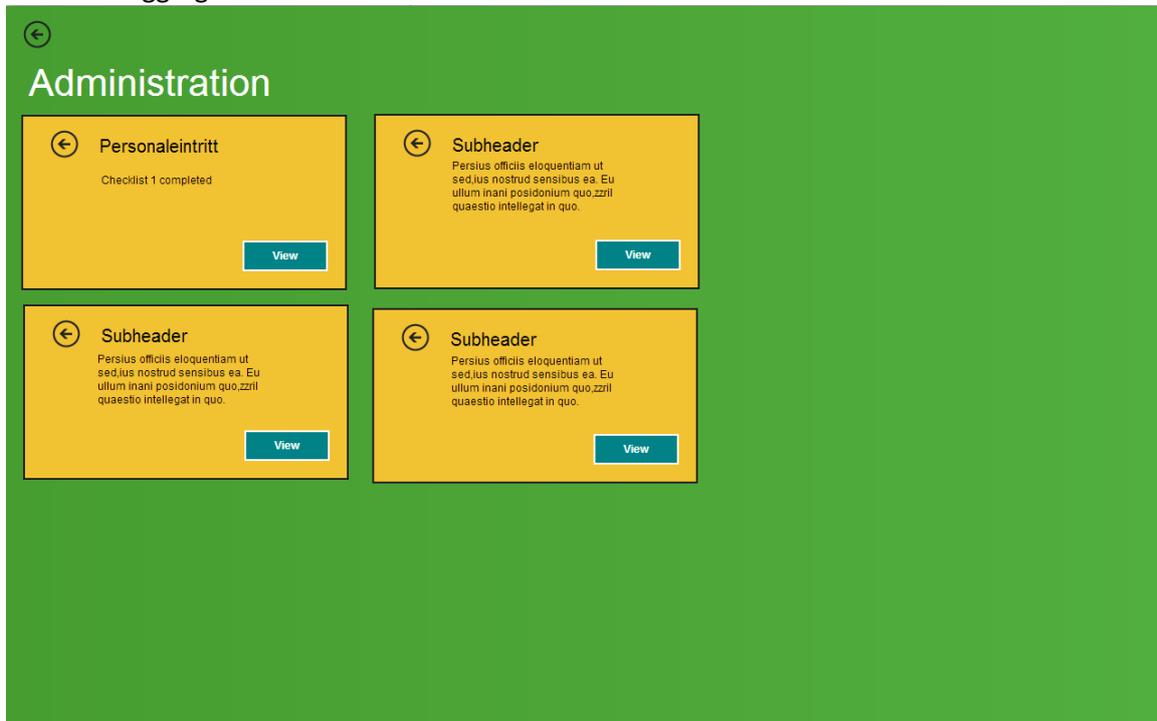


#### Mockup 5 Prozessinstanzliste

- Die Kacheln links können wiederum verschiedene Informationen anzeigen
  - Zum Beispiel: Instanzname, Status, Avatar des zugewiesenen Benutzer usw.
- Die Liste rechts zeigt alle Prozessinstanzen

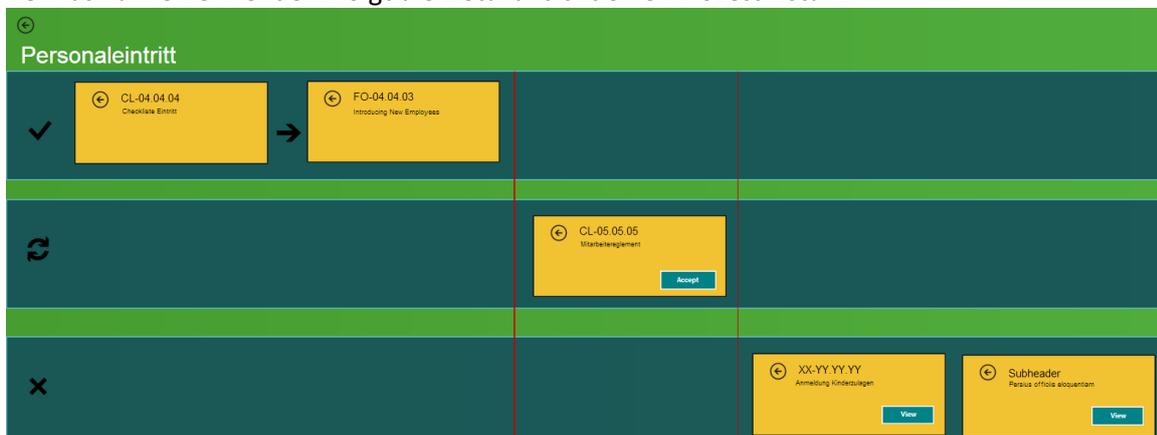
### 13.3.3.4 Administration

Bei der Administration wird eine Übersicht der anstehenden Pendenzen angezeigt. Das heisst, hier wird alles aggregiert.



#### Mockup 6 Administration

Bei Auswahl einer Pendezen folgt die Detailansicht einer Prozessinstanz.

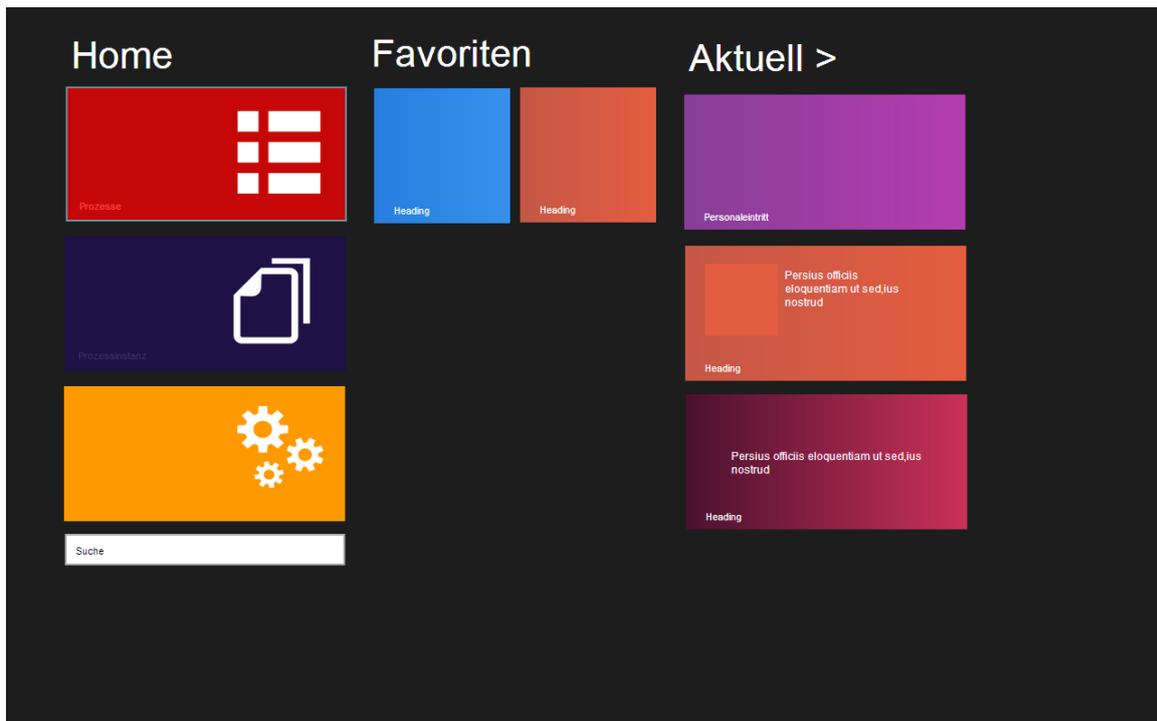


#### Mockup 7 Quittierung in Ampel-Darstellung

Es gibt hier drei verschiedene Kategorien:

- Checklisten, welche ‚completed‘ und auch vom Controller quittiert worden sind
- Checklisten, welche auf die Quittierung durch den Controller warten
- Weitere Checklisten der Prozessinstanz, welche noch nicht bearbeitet worden sind.
- Der horizontale Balken gruppiert die Kacheln und der vertikale rote Strich trennt sie.

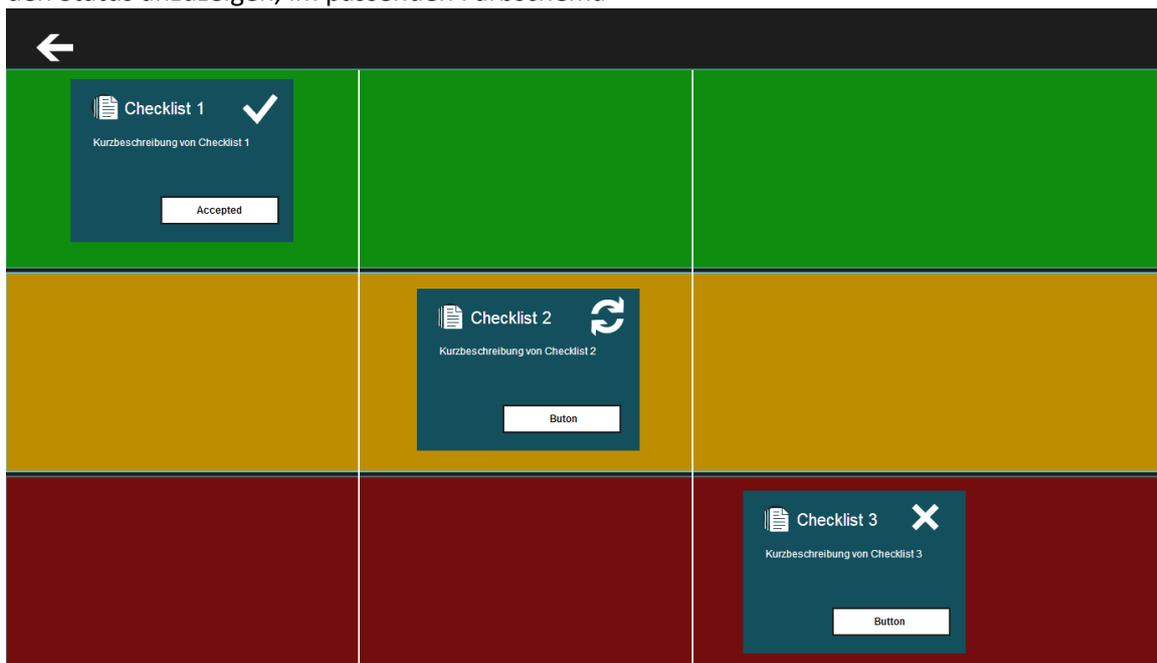
### 13.3.4 Variante 2



#### Mockup 8 Variante 2

- Hier kommt ein anderes Farbschema zum Zug
- Die linke Spalte beinhalten ‚Prozesse‘, ‚Prozessinstanzen‘ und ‚Administration‘
- Die mittlere Spalte beherbergt die benutzerdefinierten Favoriten
- Die Pendenzen werden unter ‚Aktuell‘ ganz rechts angezeigt da diese Gruppe am ehesten expandieren wird um weitere Kacheln anzuzeigen.
- Eine Suchfunktion wäre griffbereit

Der folgende Screen zeigt die Ampel-Darstellung mit den Checklisten mit einem kleinen Symbol und den Status anzuzeigen, im passenden Farbschema



#### Mockup 9 Quittierung in Ampel-Darstellung

- Das Häkchen bedeutet, diese Checkliste hat den Status ‚completed‘ und wurde vom Controller quittiert.
- Die in sich kehrenden Pfeile stellen Checklisten dar, welche den Status completed haben und von Controller quittiert werden müssen.
- Das Kreuzchen signalisiert weitere Checklisten.
- Ebenfalls wird hier auf die horizontale und vertikale Trennung zurückgegriffen.

### 13.3.5 Variante 3

Als eine auf den Controller ausgerichtete Darstellung wird die Idee des Dashboards übernommen. Wie aus diesem Beispiel welches ich im Internet gefunden habe<sup>ii</sup> zu erkennen, bekommt man auf einen Blick alles zu sehen, was relevant ist.

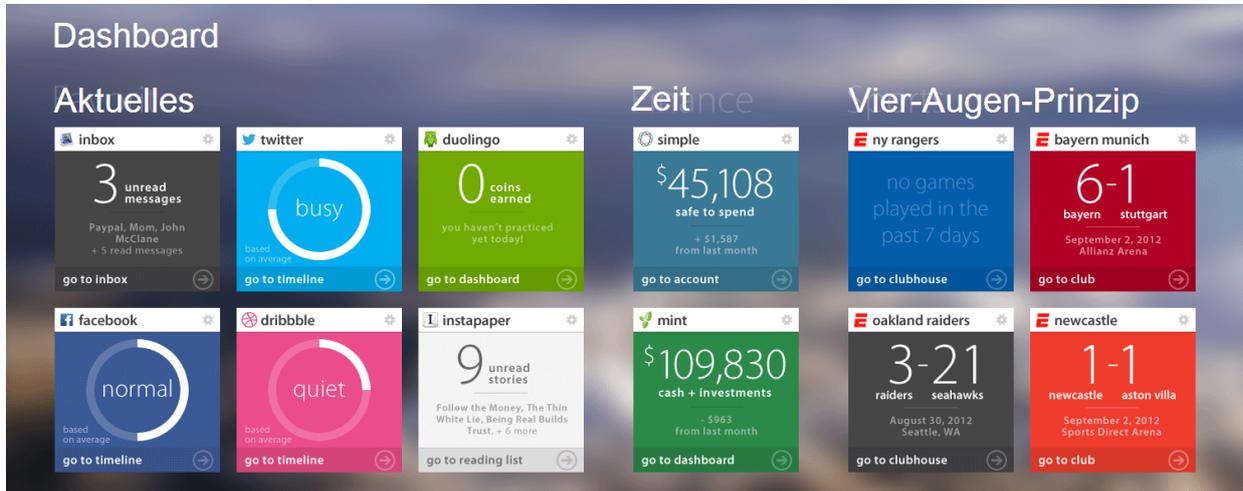


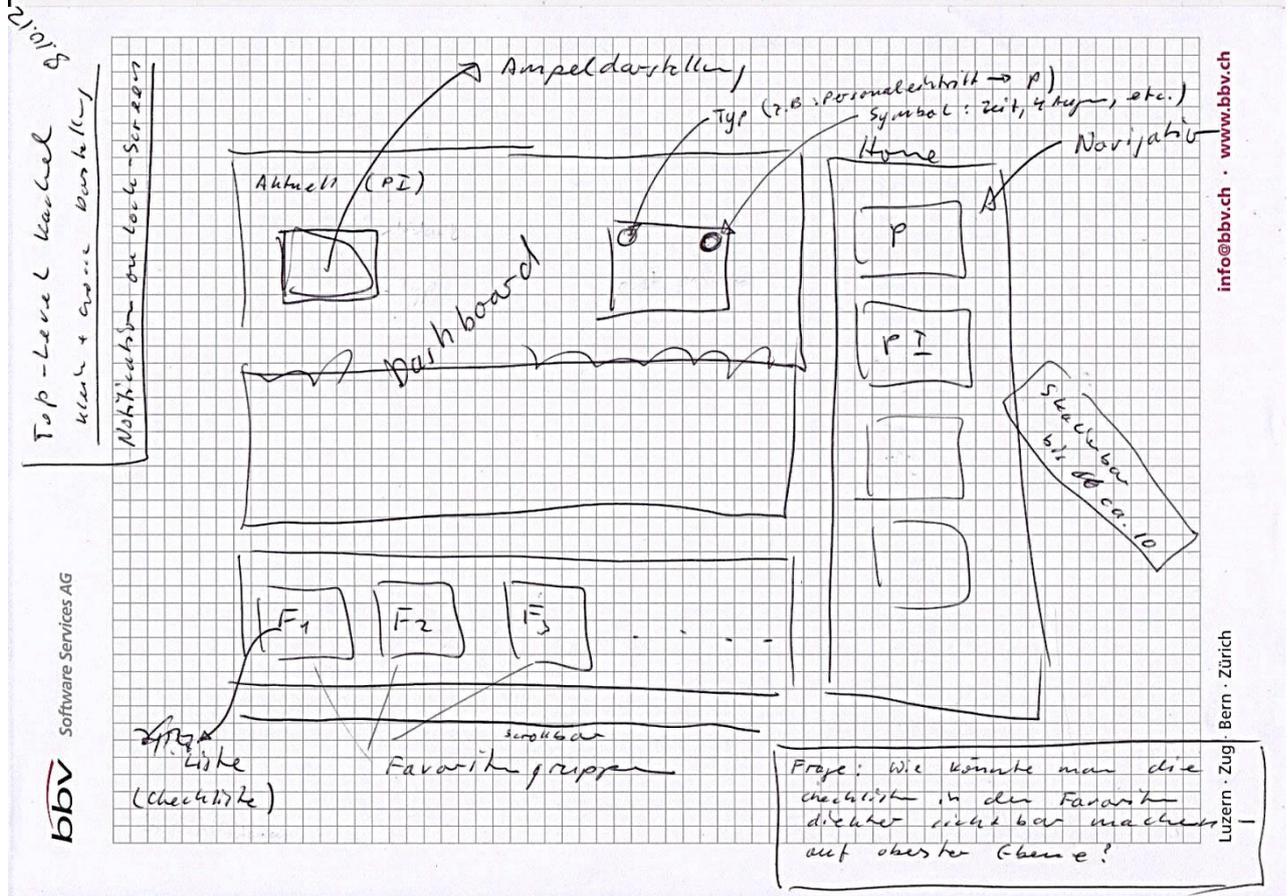
Abbildung 44 Dashboard aus dem Internet

<p><b>Pendenzen, die die grösste Aufmerksamkeit verlangen</b></p>	<p><b>Pendenzen mit ablaufenden Timer</b></p>	<p><b>Für jede weitere Person, die die gleiche Pendezen quittieren muss, wird der Avatar der Person angezeigt mit einem Statussymbol.</b></p>
---	---	---

### 13.3.6 Resultat

Kein Mockup aus dem zweiten Entwurf konnte als Lösung akzeptiert, jedoch einige Elemente als verwendbar bezeichnet werden.

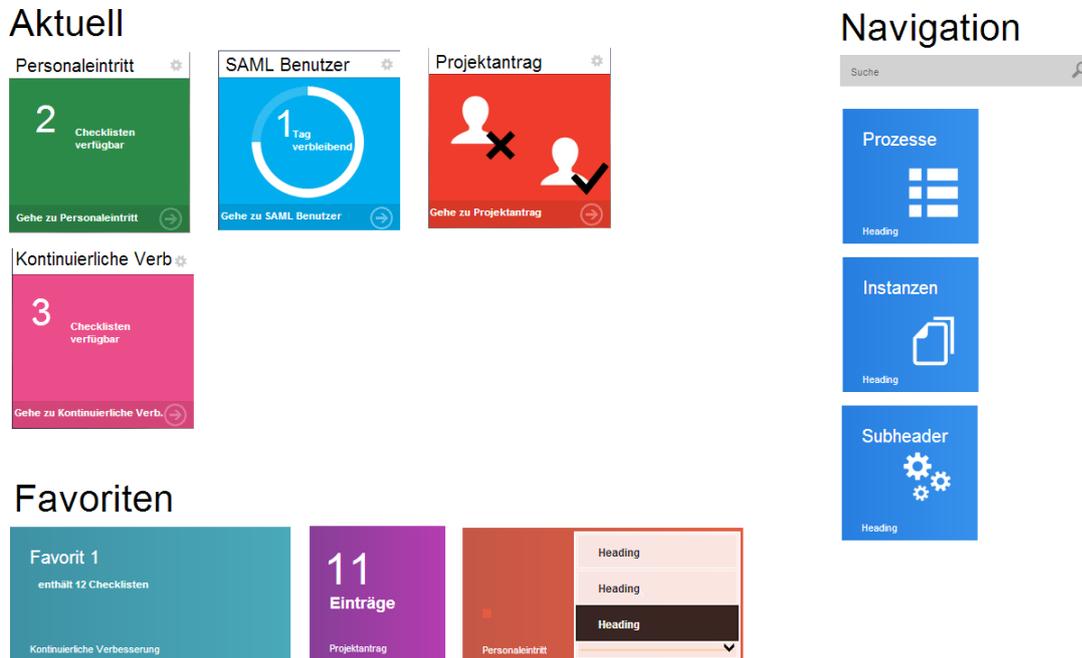
Der Auftraggeber hat basierend auf verschiedene Elemente eine Skizze erstellt:



Skizze 5 Papier-Prototyp Auftraggeber

## 13.4 Dritter Entwurf

Aufgrund der oben gegebenen Skizze wurde folgender Mockup entwickelt:



### Mockup 10 Dritter Entwurf

Dieser Entwurf wurde dem Auftraggeber samt folgenden Anmerkungen geschickt:

- Als Hintergrund wurde „light“ ausgewählt. Es handelt sich so zu sagen um den grauen Gradient, welchen Sie am Samsung Tablet gestern beobachten konnten. Der Hintergrund erschien dort als Gradient weil die Ausleuchtung des TFTs nicht optimal war. Ich konnte den gleichen Sachverhalt auch auf meinem Monitor zu Hause beobachten. Ich schlage deshalb vor, dass wir die Farbe des Hintergrunds erst nachdem wir das Ziel-Tablet in den Händen haben, definitiv festlegen.
- Die Darstellung der Avatars und die Visualisierung&Validierung werden bei der Implementierung noch verfeinert.
- In der letzten Kachel der Favoriten sehen Sie eine Liste mit den ersten beinhaltenden Checklisten. Diese kann gescrollt werden.

Das Feedback des Auftragsgebers:

1. Grundsätzlich hätte ich einen höheren Fertigstellungsgrad erwartet. Für mich wirkt es noch eher wie im Ideenstatus als ein ausgearbeiteter Vorschlag. Bitte den Entwurf fertigstellen: Also Header und Subheader mit Inhalt füllen. Legende erstellen, so dass man die Semantik der Kacheln besser versteht. Varianten/Optionen in der Legende behandeln.
2. Die Favoriten tragen aktuell die Namen der Prozesse. Ich denke wir sollten da geschicktere Bezeichner finden. Mir sind zudem die 3 Formate unklar. Falls ich auswählen kann, dann würde ich die violette Kachel für die kleine Darstellung und die letzte Kachel für grosse Darstellungen (also standardmässige Darstellung) wählen.
3. Farbkonzept: Die Navigation trägt eine einheitliche Farbe, Aktuell und Favoriten nicht. Weshalb? Hier wäre ich froh, wenn Sie sich mit einen Usability Spezialisten an Ihrer Schule austauschen, was denn ein modernes und sinnvolles Farbkonzept wäre. Ich kann hier keinen professionellen Feedback einbringen.

4. *Favoriten: Hier bin ich mir nach wie vor unklar, ob Checklisten direkt auf oberster Ebene sichtbar sein sollen, oder ob nur Gruppen oder beides möglich sein soll. Ich bitte Sie eine entsprechende Diskussion mit Ihrem Umfeld zu führen und den Trend rauszuspüren.*
5. *Aktuell: Mir ist nicht klar ob die Anzahl der Checklisten die auf den Kacheln genannt sind zum gleichen oder zu unterschiedlichen Prozessinstanzen gehören. Falls es Checklisten mehrerer Instanzen sind, dann kann ich keine klare semantische Abtrennung von den Favoriten erkenne (da werden ja Checklisten auch nach Prozessen gruppiert). Die vierte Kachel (pink) verstehe ich in der Semantik nicht – weshalb hat es kein Piktogramm?*
6. *Gehe zu ....: dies ist nicht einheitlich angewendet (nur bei Aktuell). Weshalb?*
7. *Suche: nach was sucht diese? Wo gehen die Suchresultate hin?*

Antworten und Begründungen auf das Feedback:

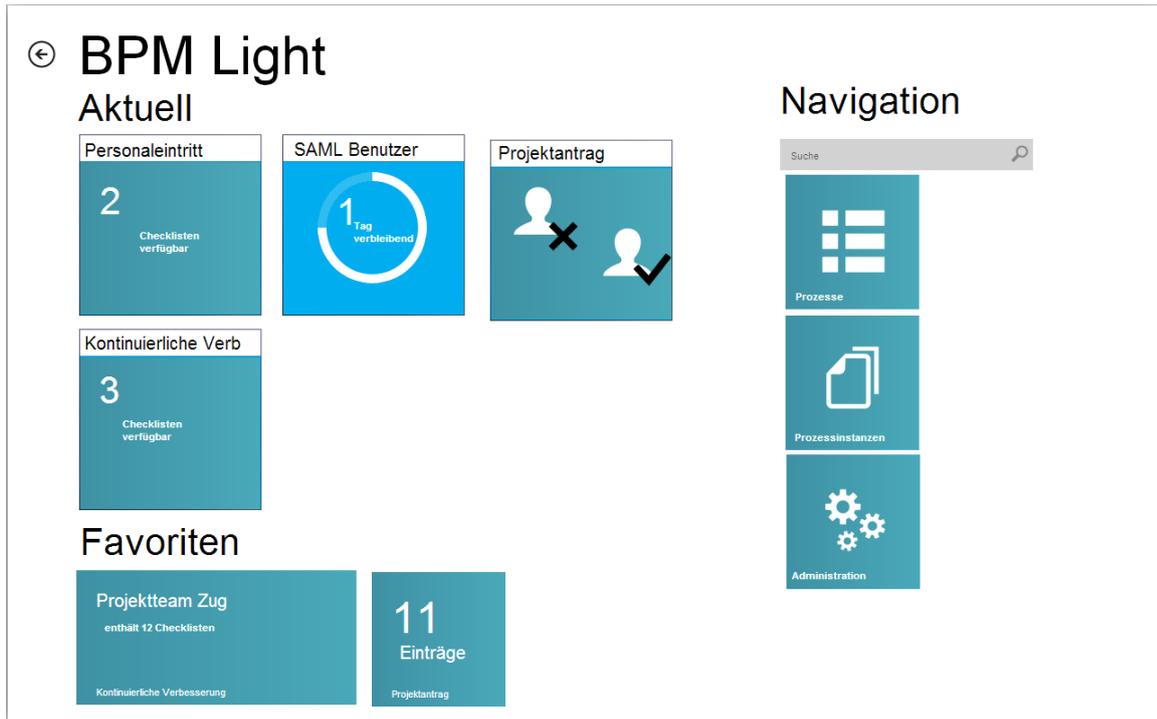
1. Das zu Verfügung stehende Mockup Tool ist in seiner Funktionen sehr eingeschränkt. Da es keine anderen Tools für die Erstellung von Mockups für Metro gibt, wäre die einzig andere Möglichkeit, das User Interface sogleich in XAML zu implementieren. Da dies sehr aufwendig ist, muss man in diesem Fall, was Chrome & Polish angeht, Abstriche machen.
2. Die Bezeichner der Favoriten sollten nur als Platzhalter verstanden werden. Der Benutzer bekommt die Möglichkeit bei Erstellung eines Favoriten die vorgegebenen Header/Subheader anzunehmen oder einen benutzerdefinierten Text einzugeben. (Bem. Der Auftraggeber hat bei einem späteren Telefongespräch den Wunsch geäußert, vollständige Freiheit zu ermöglichen.)
3. Bei einem Meeting wurde der Wunsch geäußert, dass die Applikation vielfarbig sein soll und deutlich als Metro Applikation erkannt wird. Deshalb wurden verschiedene Farben verwendet.  
Die Navigations-Kacheln habe eine einheitliche Farbe, da die Kacheln an sich definiert sind. Es sind Kacheln, um zu navigieren.  
Bei den restlichen Kacheln ist aber nie bestimmt, was für eine Prozessinstanz die Kachel beinhaltet bzw. verkörpert. Aus diesem Grund sind verschiedene Kacheln dargestellt.
4. Die Darstellung einer Liste innerhalb einer Kachel würde sie mit zu vielen Informationen überladen. Deshalb sollte auf die Liste verzichtet werden.
5. Korrekt. Für jede Instanz sollte eine eigene Kachel dargestellt werden.
6. Die Kacheln unter aktuell unterscheiden sich optisch von den Standard Metro Kacheln(Live Informationen). Um dem Benutzer dennoch zu vermitteln, das diese Kacheln weiterhin anklickbar ist, steht bei diesen Kacheln „Gehe zu...“
7. Die Windows 8 Suchfunktion bietet die Möglichkeit ein Suchwort einzugeben und dann die Applikation auszuwählen worin gesucht werden muss. Falls dies technisch kein grosser Aufwand ist, hätte ich diese Funktion gerne integriert.

### 13.4.1 Resultat

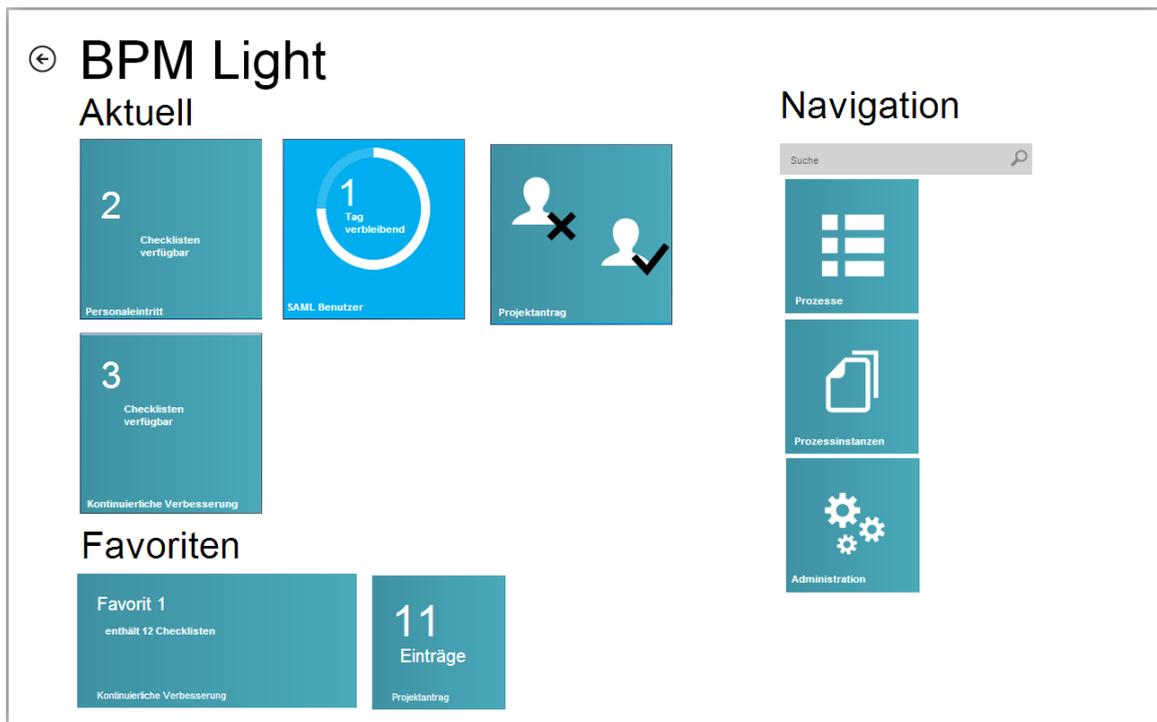
Das Mockup muss professioneller und kohärenter erstellt werden.

## 13.5 Vierter Entwurf

Aus dem Feedback und Emailverkehr mit dem Auftraggeber wurden folgende Mockup-Varianten erstellt. Sie unterscheiden sich nur darin, wie die Kacheln von Aktuell aussehen.



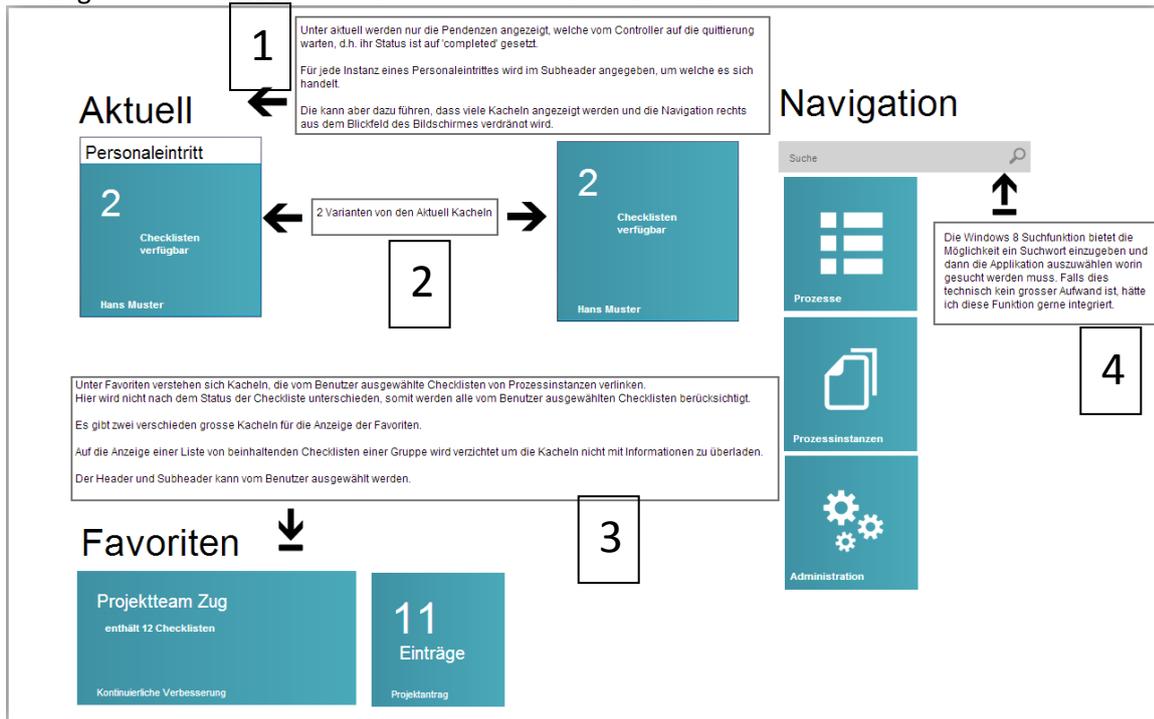
**Mockup 11 Aktuell Kachel haben einen weissen Balken mit Namen der Prozessinstanz**



**Mockup 12 Die Farbe der Kachel füllt diese ganz aus. Kein weisser Balken**

Um eine kohärente Darstellung unter den Metro Kacheln zu gewährleisten, wäre Mockup 12 eher zu empfehlen.

Ein Legende um die Semantik und die Unterschiede zu erklären:



### Legende 1

Da der Text schwer zu lesen ist, hier eine Liste:

1. Unter aktuell werden nur die Pendenzen angezeigt, welche vom Controller auf die Quittierung warten, d.h. ihr Status ist auf 'completed' gesetzt.  
Für jede Instanz eines Personaleintrittes wird im Subheader angegeben, um welche es sich handelt.  
Die kann aber dazu führen, dass viele Kacheln angezeigt werden und die Navigation rechts aus dem Blickfeld des Bildschirms verdrängt wird.
2. 2 Varianten von den Aktuell Kacheln
3. Unter Favoriten verstehen sich Kacheln, die vom Benutzer ausgewählte Checklisten von Prozessinstanzen verlinken.  
Hier wird nicht nach dem Status der Checkliste unterschieden, somit werden alle vom Benutzer ausgewählten Checklisten berücksichtigt.  
Es gibt zwei verschieden grosse Kacheln für die Anzeige der Favoriten.  
Auf die Anzeige einer Liste von beinhaltenden Checklisten einer Gruppe wird verzichtet um die Kacheln nicht mit Informationen zu überladen.  
Der Header und Subheader kann vom Benutzer ausgewählt werden.  
Prozessinstanzen statt Checklisten.
4. Die Windows 8 Suchfunktion bietet die Möglichkeit ein Suchwort einzugeben und dann die Applikation auszuwählen worin gesucht werden muss. Falls dies technisch kein grosser Aufwand ist, hätte ich diese Funktion gerne integriert.

### 13.5.1 Resultat

Die Mockups verschaffen laut Auftraggeber keine gewünschte Klarheit. Es müsse folgendes angezeigt werden:

- Was repräsentiert eine einzelne Kachel(Prozess, Prozessinstanz, Checkliste)?
- Was sind die funktionalen Unterschiede der Kacheln?
- Was bedeuten die Informationen auf den Kacheln
- Was passiert beim Öffnen einer Kachel und was wird dabei dargestellt?

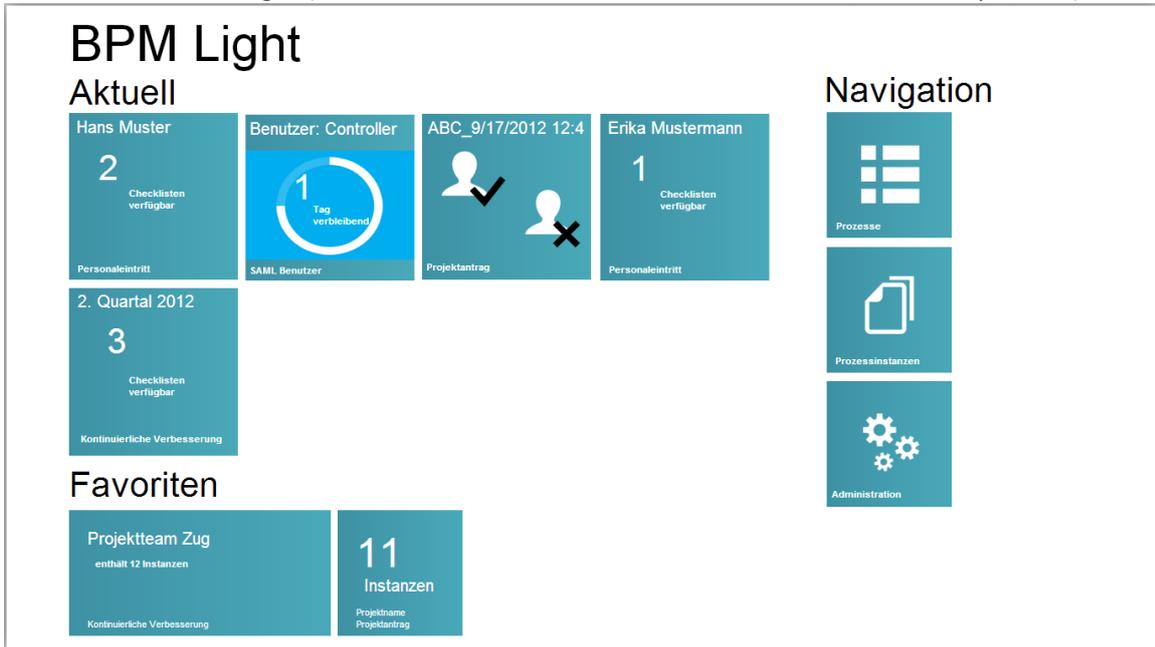
- Etc.

## 13.6 Fünfter Entwurf

Seit kurzem hat Microsoft einige Spezifikationen des Zieltablets genannt. Das wirklich relevante für dieses Projekt ist die Auflösung des Bildschirms. Diese beträgt 1366 x 768 Bildpunkte. Aus Platzgründen in diesem Dokument kann aber nicht immer gewährleistet werden, dass bei den folgenden Screenshots das Seitenverhältnis immer stimmt.

### 13.6.1 Startseite

Startseite von BPM Light (Seitenverhältnis Microsoft Surface RT Tablet, 1366x768px, 16:9)



#### Mockup 13 Startseite

##### 13.6.1.1 Aufbau

Der Aufbau lässt sich grundsätzlich in drei Gruppen unterteilen. Aktuell, Favoriten und Navigation.

##### 13.6.1.1.1 Aktuell

Der Fokus des Benutzers sollte hierauf geleitet werden.

Unter aktuell werden drei Typen von Kacheln angezeigt:

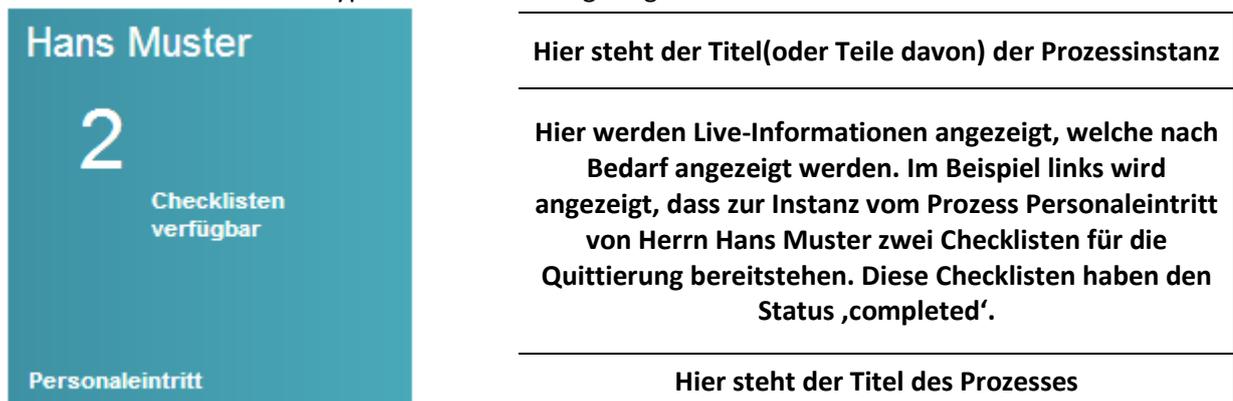


Abbildung 45 normale Aktuell-Kachel



Abbildung 46 Timer

Hier steht der Titel der Prozessinstanz
Einige Checklisten müssen innerhalb einer Zeit abgearbeitet werden. Deswegen wird bei der Kachel ein ablaufender Timer angezeigt um die verbleibende Zeit anzuzeigen. Die Grafik wird per Live-Information aktuell gehalten. (Die Farbe des Timers soll hier vernachlässigt werden)
Hier steht der Titel des Prozesses



Abbildung 47 Vier-Augen-Prinzip

Hier steht der Titel der Prozessinstanz
Einige Checklisten müssen durch das Vier-Augen-Prinzip quittiert werden. Durch den Avatar (falls vorhanden) kann signalisiert werden, von wem die Checkliste schon quittiert worden ist. Falls kein Avatar verfügbar ist, wird der Name der Person angezeigt.
Hier steht der Titel des Prozesses

### 13.6.1.1.2 Favoriten

Favoriten werden vom Benutzer erstellt. Er kann hier eine Menge von Prozessinstanzen eines Prozesses auswählen, welche er des Öfteren beobachten will. Bei Betrachtung einer Instanz aus einem Favoriten wird eine Ampel-Darstellung (wird weiter unten gezeigt) verwendet um den Status der beinhaltenden Checklisten anzuzeigen.

Diese Kacheln können in zwei verschiedenen Grössen angezeigt werden (310x150px und 150x150px, standardmässige Grösse von Metro Kacheln), wobei sie den gleichen Informationsgehalt bieten können. Der Benutzer bekommt die Möglichkeit, den Titel selber zu definieren (im Beispiel: ‚Projektteam Zug‘).

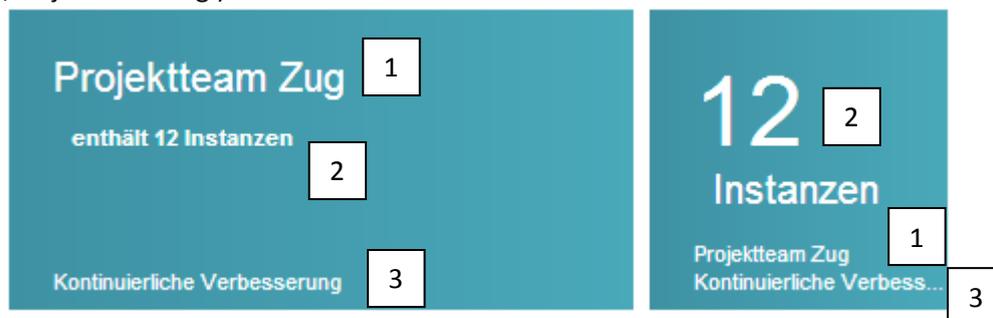


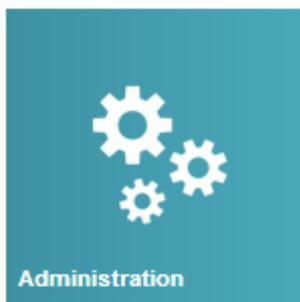
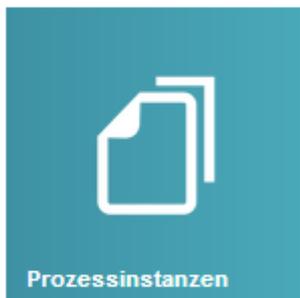
Abbildung 48 Grosse und kleine Favoriten-Kachel

4. Vom Benutzer ausgewählter Name des Favoriten

5. Live-Information mit der Anzahl der beinhaltenden Prozessinstanzen
6. Name des Prozesses

### 13.6.1.1.3 Navigation

## Navigation

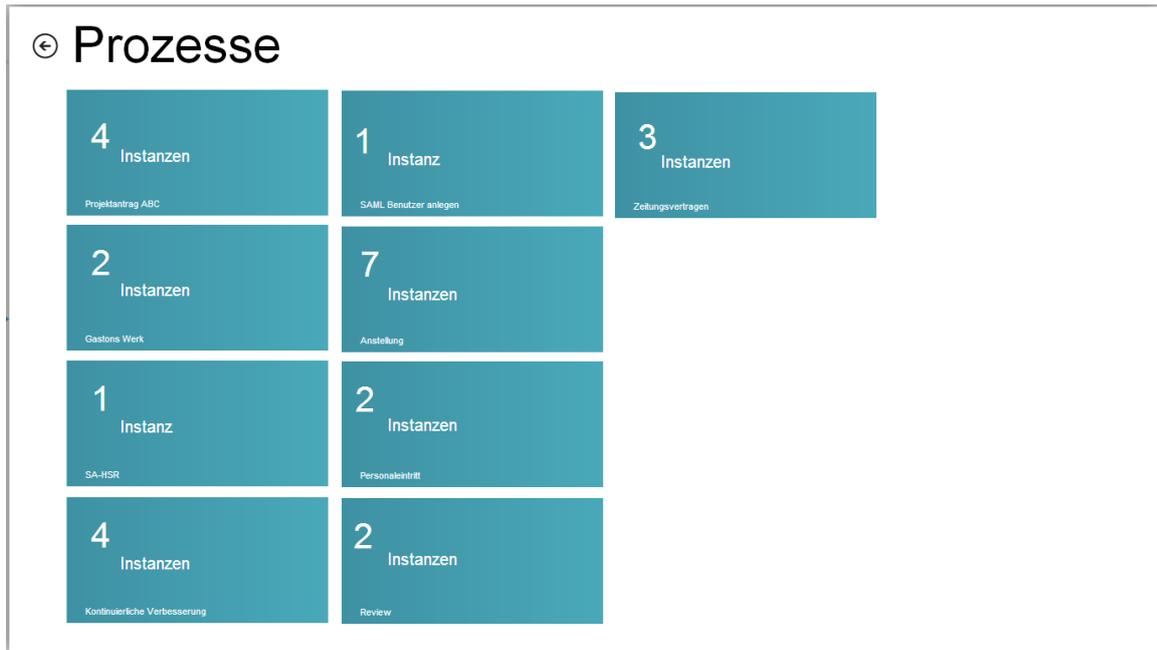


<b>Link zu allen Prozessen</b>
<b>Link zu allen Prozessinstanzen</b>
<b>Soweit ist diese Kachel noch nicht definiert. In der SharePoint-Implementation gibt es hier einen Logviewer. Diesen in der Metro-Applikation bereit zu stellen, ist abzuraten.</b>

Abbildung 49 Navigations-Kacheln

### 13.6.2 Prozesse

Jeder Prozess wird als eine eigenständige Kachel dargestellt und nicht als eine vertikal scrollbare Liste, da die Kacheln besser zum ganzen Gefüge passen und den Platz des Bildschirmes verwerten.



**Mockup 14 Alle Prozesse**

Detail-Beschreibung einer Kachel:

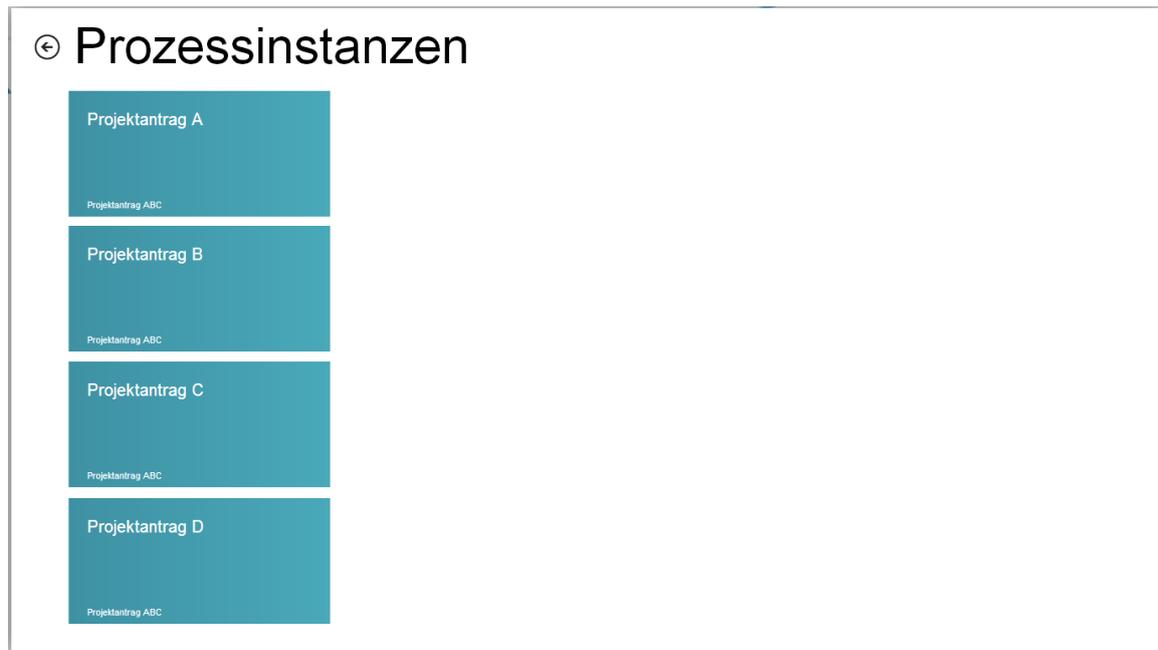


**Abbildung 50 Grossansicht Prozess-Kachel**

Jede dieser Kacheln ist anklickbar und führt zur Detailansicht (Kapitel 2.4.4 auf Seite 14).

### 13.6.3 Prozessinstanzen

Hier wird die gleiche Idee verfolgt, wie bei der Darstellung von den Prozessen.



Mockup 15 Alle Prozessinstanzen



Abbildung 51 Grossansicht Prozessinstanz-Kachel

### 13.6.4 Detailansicht

Um wiederverwertbare Elemente zu erstellen, bietet sich bei der Detailansicht eine Master/DetailView zu verwenden an. Darin können sowohl die Liste der Prozessinstanzen als auch die Zustände der Checklisten der zugehörigen Prozessinstanz angezeigt werden.

Unter dem Kapitel 13.3.3.4

Administration auf Seite 60 unter Mockup 7 Quittierung in Ampel-Darstellung wird die Bedeutung von den drei horizontalen Balken, welche unten in Abbildung 20 mit der Nummer 3 dargestellt sind, erklärt. Hier eine Wiederholung davon:

Es gibt hier drei verschiedene Kategorien:

- 3a) Checklisten, welche ‚completed‘ und auch vom Controller quittiert worden sind
  - 3b) Checklisten, welche auf die Quittierung durch den Controller warten
  - 3c) Weitere Checklisten der Prozessinstanz, welche noch nicht bearbeitet worden sind.
- Der horizontale Balken gruppert die Kacheln und der vertikale rote Strich trennt sie.

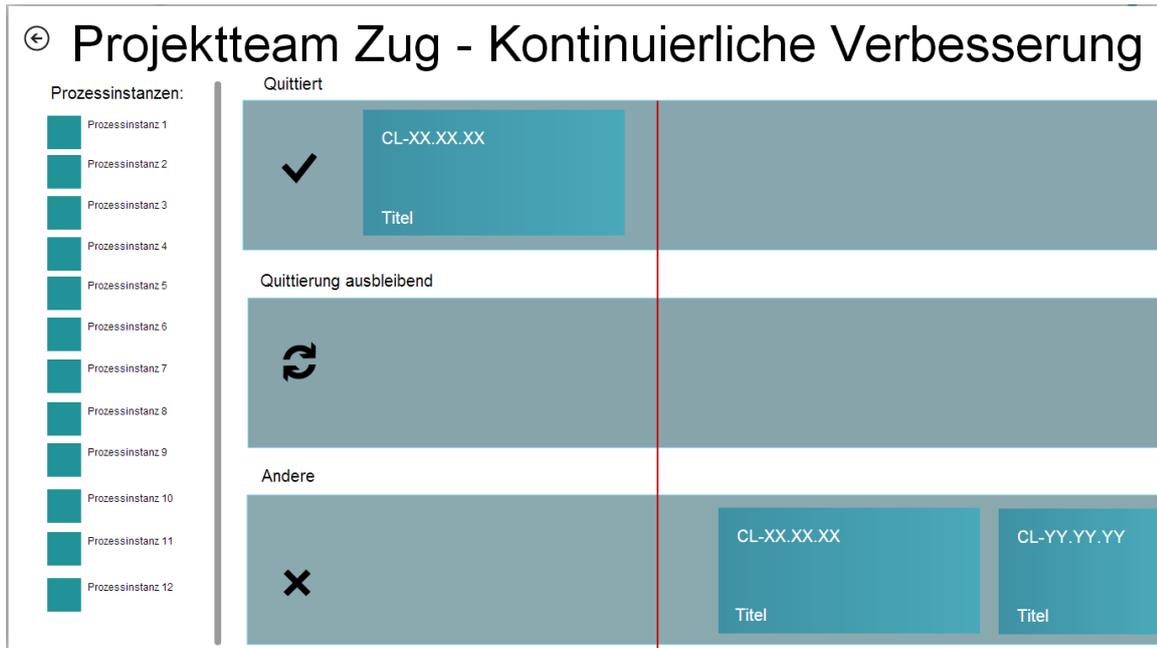


### Mockup 16 Master/Detail-Ansicht

4. Name des Prozesses
5. Liste der Prozessinstanzen
6. Darstellung der Checklisten einer ausgewählten Prozessinstanz (welche Prozessinstanz links ausgewählt ist, kann entweder durch farbliches hervorheben oder durch das Hinzufügen des Namen der Instanz im Titel [Nr.1 im Bild] angezeigt werden.)
  - a. Quittierte Checklisten
  - b. Zur Quittierung bereitstehenden Checklisten
  - c. Weitere Checklisten

#### 13.6.4.1 Favoriten

Die Darstellung eines Favoriten bedient sich an der Master/Detail-Ansicht Konzept. Folgender Mockup wird hier gezeigt, da der Favorit direkt aus der Startseite (Abbildung 10 Startseite) aufgerufen werden kann.



Mockup 17 Favoriten Darstellung

### 13.6.4.2 Checkliste quittieren

Als nächstes wird beschrieben, wie eine Quittierung abläuft wenn man diese direkt aus der Startseite aufgreift. Das Prinzip wird auch bei der Master/Detail-Ansicht verwendet.

4. Der Benutzer sieht, dass im mittleren horizontalen Balken eine Checkliste zum Quittieren bereit steht:



Mockup 18 Checkliste quittieren

5. Der Benutzer klickt die entsprechende Kachel. Es wird im Abbildung 23 gezeigt:

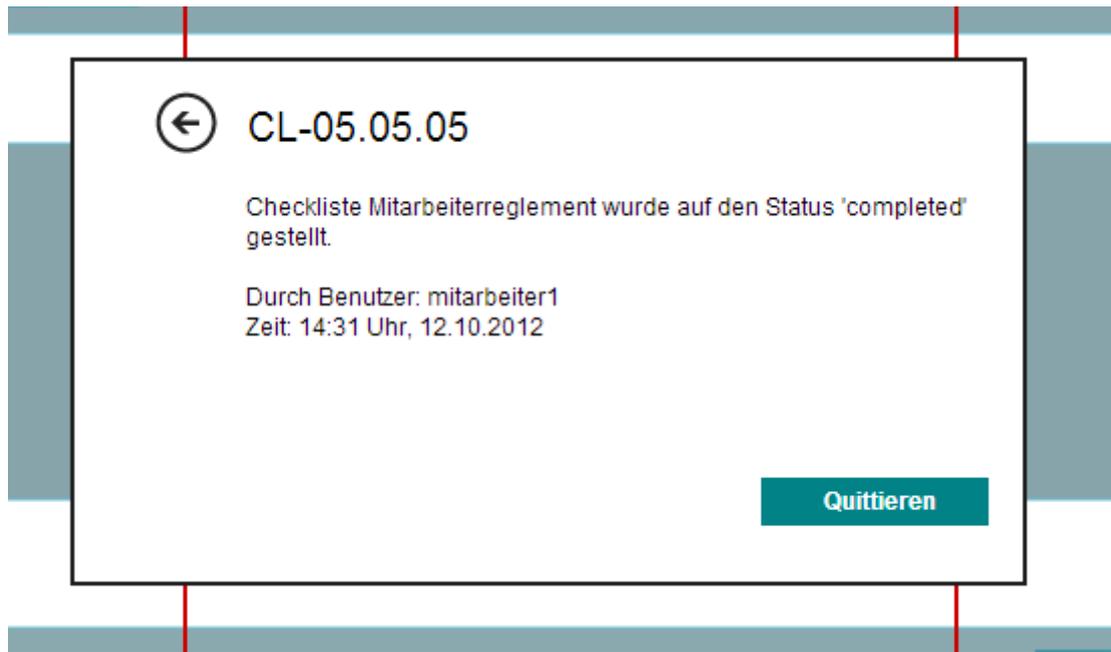


Abbildung 52 Quittierung

6. Der Benutzer quittiert die Checkliste.  
Hier wird angezeigt, durch welchen Benutzer und wann diese Checkliste auf den Status ‚completed‘ gestellt wurde.  
Die quittierte Kachel rutscht einen Balken nach oben:



Mockup 19 Quittierung durchgeführt

### 13.6.5 Resultat

Ich hoffe, die Fragen des Auftraggebers konnte somit beantwortet werden. Eine Wiederholung der Frage:

- Was repräsentiert eine einzelne Kachel(Prozess, Prozessinstanz, Checkliste)?
- Was sind die funktionalen Unterschiede der Kacheln?
- Was bedeuten die Informationen auf den Kacheln
- Was passiert beim Öffnen einer Kachel und was wird dabei dargestellt?

Eine Sammlung von den Mockups wird in Zukunft im Repository verfügbar sein.

<sup>i</sup> Mockups-Tool [www.fluidui.com](http://www.fluidui.com)

<sup>ii</sup> Dashboard Mockup <http://dribbble.com/shots/714820-Metro-Dashboard> (Stand 17.10.2012)